NPS67-85-012

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

REVIEW, IMPLEMENTATION AND TEST OF THE QAZ1D COMPUTATIONAL METHOD WITH A VIEW TO WAVE ROTOR APPLICATIONS

by

Thomas F. Salacka

December 1985

Thesis Advisor: Ray P. Shreeve

Prepared for: Naval Air Systems Command
Washington, DC 20361

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Robert H. Shumaker
Superintendent

David A. Schrady
Provost

This thesis prepared in conjunction with research sponsored in part by the Naval Air Systems Command, Washington, DC, under Air Task #A3l03l0E/l86A/WR024-03-00l.

Released By:

UNCLASSIFIED
URITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
DECLASSIFICATION / DOWNGRADING SCHEDULE	
ERFORMING ORGANIZATION REPORT NUMBER(S) PS67-85-012	S. MONITORING ORGANIZATION REPORT NUMBER(S) NPS67-85-012
NAME OF PERFORMING ORGANIZATION val Postgraduate School 6b. OFFICE SYMBOL (If applicable) Code 67	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
ADDRESS (City, State, and ZIP Code) nterey, California 93943-5100	7b. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5100
NAME OF FUNDING / SPONSORING ORGANIZATION val Air Systems Command 8b. OFFICE SYMBOL (If applicable) Air 310E	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N0001985WR 51147
ADDRESS (City, State, and ZIP Code) ashington, D.C. 20361	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 61153N; PROJECT NO: WR024; TASK NO: 03; WORK UNIT ACCESSION NO: 001

TITLE (Include Security Classification)
EVIEW, IMPLEMENTATION AND TEST OF THE QAZ1D COMPUTATIONAL METHOD WITH A EW TO WAVE ROTOR APPLICATIONS

PERSONAL AUTHOR(S)
lacka, Thomas F.

TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
ter's Thesis	FROM ____ TO ____	1985, December	119

SUPPLEMENTARY NOTATION

COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	QAZ1D; Wave Rotor; Computational Fluid Dynamics; Euler

ABSTRACT (Continue on reverse if necessary and identify by block number)

The QAZ1D method is redeveloped in detail and implemented in a first order, one-dimensional FORTRAN program, EULER-1. The program is tested on the shock tube problem and results are presented for various computational meshes and initial conditions. Based on good results of the EULER-1 code, recommendations are made for future

DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
NAME OF RESPONSIBLE INDIVIDUAL Prof. Ray P. Shreeve	22b TELEPHONE (Include Area Code) (408) 646-2593 22c OFFICE SYMBOL Code 67Sf

FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

#19 - ABSTRACT - (CONTINUED)

extensions and testing to validate the suitability of the QAZ1D method for wave rotor applications.

Approved for public release; distribution is unlimited.

Review, Implementation and Test of the QAZ1D Computational Method with a View to Wave Rotor Applications

by

Thomas Francis Salacka
Lieutenant, United States Navy
B.S., United States Naval Academy, 1977

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1985

ABSTRACT

The QAZ1D method is redeveloped in detail and implemented in a first order, one-dimensional FORTRAN program, EULER-1. The program is tested on the shock tube problem and results are presented for various computational meshes and initial conditions. Based on good results of the EULER-1 code, recommendations are made for future extensions and testing to validate the suitability of the QAZ1D method for wave rotor applications.

TABLE OF CONTENTS

LIST OF FIGURES

TABLE OF SYMBOLS

Below are two tables of symbols. The first table lists all the symbols which occur in the text and, where applicable, their FORTRAN counterparts used in the EULER-1 code. The second table lists the remaining symbols used in the EULER-1 code which were not listed in the first table. Both tables are in alphabetical order.

SYMBOLS USED IN THE TEXT

Text	EULER-1	Definition
A	A	Speed of sound
P	PRESS	Static pressure
Q	QQ	Modified Riemann variable ($Q = q + AS$)
q	Q	Absolute velocity magnitude
Q_R		Reversible heat transferred
R	RR	Modified Riemann variable ($R = q - AS$)
R_G		Gas constant
S	S	A modified form of entropy
$\hat{\imath}_{s,n,m}$		Unit vectors in the s, n, and m directions
T	TEMP	Static Temperature
t	T	Time
u	U	Velocity magnitude relative to a steady shock wave
V_s	VS	Shock wave velocity
W	W	Mach number relative to a steady shock wave

$\bar{w}$		Vector of the principle variables, Q,R,S
$\bar{Z}$	Z(K)	Vector of the right hand sides of the governing equations
λ	LMD(K)	The characteristic trajectories in the space-time plane (q+A, q-A, q)
Δ		Small spatial change
δ		Small change with respect to time
θ, ϕ		Flow angles with respect to reference coordinate planes
ρ	DENS	Density
γ	G	Ratio of specific heats

ADDITIONAL SYMBOLS USED IN EULER-1

Symbol	Definition
*A	Suffix which denotes the left side of a discontinuity (* may be any variable)
ABAR(K)	The average value of A between the limits of integration of Z(K)
*B	Suffix which denotes the right side of a discontinuity (* may be any variable)
AR	The ratio of the sound speeds across a shock, (high/low)
COUNT	A counter for the number of time steps
DARRAY	The array of density values to be plotted by the graphics routines
DEL**H	The change in the variable ** in going from node I to node I+1 [**(I+L) - **(I)]
DELX(K)	The distance from the point where the K^{th} characteristic crosses the known time level to the I^{th} node, measured positive to the right
DEL**L	The change in the variable ** in going from node I-1 to node I [**(I) - **(I-1)]

DLCD	The exact value of the density to the left of a right propagating contact discontinuity in the Riemann problem
DLSH	The exact value of the density to the left of a right propagating shock in the Riemann problem
DLTA**	A prefix which indicates the spatial change in ** over one time step
DQ	The change in the velocity magnitude across a shock, (high-low)
DR	The ratio of the density across a shock, (low/high)
EE	A value which indicates the precision to which the characteristics are to be calculated
E(K)	The computed error in the calculation of the characteristics
GRAPHS	Parameter which controls the type of output produced by EULER-1. 0 = Tabular, 1 = Plot of Q, S, pressure and density, 2 = Comparison of density with exact solution
G1	1/(G-1)
G2	2/(G-1)
H	The non-dimensional value of one spatial interval 1/(N-1)
I	Subscript which denotes the spatial node
**INT(K)	The interpolated value of ** at the point on the known time level where the K^{th} characteristic crosses
INTEG(K)	The result of integrating Z(K)
I2(L)	The node to the right of discontinuity L
J	The time level
JSTOP	The number of time levels desired to be computed
K	A subscript which indicates the characteristic being dealt with. 1 = Q+A 2 = Q-A 3 = Q

L	A subscript which denotes the type of discontinuity. 1 = schock 2 = contact discontinuity 3 = head of rarefraction wave 4 = tail or rarefraction wave
M	Parameter which controls the types of discontinuities which EULER-1 tracks. 1 = shocks only. 2 = shocks and contact discontinuities
N	The number of nodes in the computational grid
NEW**	A temporary storage location for variables ** updated in time
PARRAY	The array of densities plotted by the graphics routines
PR	The ratio of pressures across a shock, (high/low)
PRI	The initial pressure ratio across the diaphragm (high/low)
*PRIM(K)	Suffix which indicates the spatial derivative of * at the present time level, where * is either A or Q
QARRAY	The array of velocities to be plotted by the graphics routines
QLI	The initial value of the velocity to the left of the diaphragm
QRI	The initial value of the velocity to the right of the diaphragm
QQJO	The measured change in the Riemann variable QQ across a shock (high-low)
QQJE	The change in the Riemann variable QQ across a shock calculated analytically (high-low)
SIGMA(L,J)	Discontinuity locations. J = 1 indicates the known time level and J = 2 indicates the unknown time level
**STEP	The change in the primary variables with respect to time during one time step (** = QQ, RR, S)

SARRAY	The array of entropy values to be plotted by the graphics routines
SKIP	The number of time steps between outputs
TRI	The initial temperature ratio across the diaphragm (high/low)
VCDE	The exact velocity of the contact discontinuity
VHEAD	The exact velocity of the head of the rarefraction wave
VTAIL	The exact velocity of the tail of the rarefraction wave
VS	The shock wave velocity computed by EULER-1
VSE	The exact velocity of the shock wave
X	The non-dimensional spatial position
XARRAY	The array of node locations used by the graphics routines
XINIT	The initial location of the discontinuities. Used to calculate the exact solution
X2(L)	The position of the node to the right of a discontinuity

ACKNOWLEDGMENT

Thanks to my wife, Kathy, for her support and encouragement, to Augie Verhoff, for his personal assistance, to Atul Mathur, for his availability and assistance, and to Ray Shreeve, for making this an enjoyable and worthwhile learning experience.

I. INTRODUCTION

A program is underway at the Naval Postgraduate School's (NPS) Turbopropulsion Laboratory to evaluate the wave rotor concept. The wave rotor, operating as a component in a gas turbine engine, uses unsteady wave propagation in tube-like passages to compress incoming air before it goes to the combustion chamber. The combustion chamber output is routed back to the rotor to create the unsteady waves. Thus the rather simple rotor, with "partial admission" inlet and outlet ports, acts both as a compressor and as a turbine. The alternation of hot and cool gases through the same hardware aids cooling and allows higher operating (cycle) temperatures to be used. The interaction of the hot and cool gases within the passages of the wave rotor through the wave patterns that are created pose a difficult problem. The work underway aims to develop and validate preliminary design and performance analysis tools.

One of the tools neededis a computer program which can be used to construct wave interactions in a design process and then accurately predict the performance of the design. Until efficient and accurate methods of solving the full Navier-Stokes equations for unsteady turbulent flows are developed, the design program will be based initially on the solution of the unsteady Euler equations and a method devised to represent losses. Once the program is developed, it must

be verified against experiment. This is the overall goal of the present program in which a wave rotor apparatus has been assembled and methods of measuring the unsteady pressures and temperatures are being developed concurrently with the computational effort.

Three different approaches to the solution of the unsteady Euler equations were examined in the overall program. First, Eidelman developed a two-dimensional code based on the Godunov method of solution [Ref. 1] and applied the code to examine unsteady wave propagation in ducts [Ref. 2] and the process of port opening to wave rotor passages [Ref. 3]. A summary of Eidelman's work is given in Reference 4. While the method is conservative and does not require the introduction of artificial viscosity, the extension from one to two dimensions is not rigorous when shock waves are present, and computational times with the present code are quite long. The extension to include viscous effects would require a separate treatment of the boundary layer.

Second, Mathur developed a one-dimensional code based on the Random Choice Method (RCM) of solution [Ref. 5]. Somewhat similar to the Godunov method, in that the solution is based on solving the Riemann problem within each grid cell at each time step, the RCM approach results in very sharp discontinuities which can be tracked easily. This is particularly useful in constructing wave rotor cycles, in which the position of the gas-gas interface is equally as important as the position

of the compression and expansion waves. The code is therefore valuable in the preliminary design process, to examine suitable port arrangements, and the gas properties at the ports, for a given task. Unfortunately, an extension to two-dimensional and/or viscous flow can not be made rigorously.

The third approach was followed in the present work. The QAZ1D method for compressible inviscid flow computations developed by Verhoff and O'Neil [Ref. 6] was implemented to generate a one-dimensional unsteady Euler code with the goal of evaluating the suitability of the approach for wave rotor applications. Some advantages of the QAZ1D method were recognized as being the following:

1. The method is based on the use of characteristics. Such methods can model wave propagation accurately.

2. The use of a natural streamline coordinate system eases the difficult task of computing with two and three dimensional grids.

3. The equations are written in a form which allows a straightforward extension to viscous flows.

4. Codes for computing internal, steady, two-dimensional flows in the presence of shocks, and simple internal viscous flows, have been generated quickly without significant development problems [Ref. 6].

In the work reported herein, a one-dimensional FORTRAN code based on the QAZ1D method was developed using the NPS IBM370-3033 computer and subsequently exercised on the shock tube test problem. In reporting the work, first, in Section II and Appendix A, a complete account is given of the derivation and non-dimensionalization of the governing equations.

In Section III, a FORTRAN code, EULER-1, is described. Its operation on the NPS computer, and the listing of the code, are given in Appendix B and Appendix C, respectively. Results of applying the code to the shock tube test problem are given in Section IV. Difficulties encountered in the implementation of the method and additional comments are given in Section V, and conclusions are given in Section VI.

II. THE QAZ1D METHOD

A. OVERVIEW

The QAZ1D method uses Riemann-like variables with a modified entropy term in their definitions, to express the Euler equations in a natural streamline coordinate system. The resulting partial differential equations (PDE), when cast along the characteristic trajectories in the space-time domain, reduce to a system of ordinary differential equations (ODE) which may be solved explicitly. The advantage in using the modified Riemann variables is that they are less affected by discontinuities in the flow than are the standard Riemann variables. Since the equations are not valid across discontinuities which cause irreversible losses, such discontinuities must be located and treated with special logic in the numerical formulation.

In the following presentation, the reader's familiarity with the method of characteristics is assumed.

B. DEVELOPMENT OF THE EQUATIONS

This section presents the governing equations and outlines their development. A rigorous derivation of the equations is presented in Appendix A.

1. The Coordinate System

The natural streamline coordinate system (s,n,m), is shown in Fig. A-1 relative to a fixed rectangular cartesian

system (x,y,z). The (s,n,m) system is a right-hand orthogonal system which undergoes curvilinear translation as it moves with a fluid particle along a streamline. The coordinate system is described in detail in Appendix A.

2. Variables

The modified Riemann variables, or "extended Riemann variables" [Ref. 6:p. 1], are defined as

$$\begin{aligned} Q &= q + AS \\ R &= q - AS \end{aligned} \tag{1}$$

where q is the velocity magnitude, A is the speed of sound and S is the modified entropy defined in terms of pressure and density as

$$S = \frac{R_G}{\gamma(\gamma-1)}[2\gamma - \ln(P/\rho^{\gamma})] \tag{2}$$

The modified entropy relation is the result of defining the entropy change, dS, to be given by

$$dS = -\frac{1}{\gamma}\frac{dQ_R}{T} \tag{3}$$

where dQ_R is the heat required to be added in a reversible process between the same end states, T is the temperature and γ is the ratio of specific heats.

3. Conservation of Mass

By applying the continuity equation to a differential stream tube of variable cross sectional area, in natural coordinates, and ignoring all third order and higher derivatives, one obtains

$$\frac{\partial \rho}{\partial t} + q \frac{\partial \rho}{\partial s} + \rho \frac{\partial q}{\partial s} + \rho q \left[\frac{\partial \theta}{\partial n} + \cos \theta \frac{\partial \phi}{\partial m}\right] = 0 \qquad (4)$$

where q is the velocity, ρ is the density, s is the streamwise spatial dimension, and θ and ϕ are the flow angles as defined in Fig. A-1 of Appendix A.

4. Conservation of Momentum

Applying the vector form of the momentum conservation law in natural coordinates, the equation of motion for inviscid flow becomes

$$\hat{i}_s \left[\rho \frac{\partial q}{\partial t} + \rho q \frac{\partial q}{\partial s} + \frac{\partial P}{\partial s}\right] + \hat{i}_n \left[\rho q \frac{\partial \theta}{\partial t} + \rho q^2 \frac{\partial \theta}{\partial s} + \frac{\partial P}{\partial n}\right] + \hat{i}_m \left[\rho q \cos \theta \frac{\partial \phi}{\partial t} + \rho q^2 \cos \theta \frac{\partial \phi}{\partial s} + \frac{\partial P}{\partial m}\right] = 0 \qquad (5)$$

5. The Conservation of Energy

In the absence of friction and heat conduction, and outside of discontinuities (through which irreversible changes consistent with mass, momentum and energy conservation are permitted), energy conservation is equivalent to the

statement that the entropy of a fluid particle does not change. In the natural coordinate system, therefore

$$\frac{dS}{dt} = \frac{\partial S}{\partial t} + q \frac{\partial S}{\partial s} = 0 \qquad (6)$$

Equation (6) will not be valid across shock waves or contact surfaces between gases having different states.

6. Transformation to a Useful Form

Equations (4) and (5) are first expressed in terms of the primary variables q, A, S, and the flow angles. Using the definition of sound speed in a perfect gas

$$A^2 = \gamma P/\rho \qquad (7)$$

Eq. (4) becomes

$$\frac{\partial A}{\partial t} + q \frac{\partial A}{\partial s} + \frac{\gamma-1}{2} A \frac{\partial q}{\partial s} + q A \frac{\gamma-1}{2}\left[\frac{\partial \theta}{\partial n} + \cos\theta \frac{\partial \phi}{\partial m}\right] = 0 \qquad (8)$$

With Eq. (3), the equation of state for a perfect gas, and the first law of thermodynamics, Eq. (5) gives

$$\frac{\partial q}{\partial t} + q \frac{\partial q}{\partial s} + \frac{2A}{\gamma-1} \frac{\partial A}{\partial s} + A^2 \frac{\partial S}{\partial s} = 0$$

$$\frac{\partial \theta}{\partial t} + q \frac{\partial \theta}{\partial s} = - \frac{A^2}{\gamma q} \frac{\partial \ell n\ P}{\partial n} \qquad (9)$$

$$\frac{\partial \phi}{\partial t} + q \frac{\partial \phi}{\partial s} = - \frac{A^2}{\gamma q \cos\theta} \frac{\partial \ell n\ P}{\partial m}$$

Equations (6), (8) and (9) constitute the system of P.D.E.'s which describe the unsteady isentropic flow of a perfect gas in natural coordinates. Since third order terms were neglected in the conservation of mass, the system is only accurate to second order.

The system of P.D.E.'s is now transformed into a system of O.D.E.'s along the characteristic trajectories. In general, if w is a function of (s,t), then

$$dw = \frac{\partial w}{\partial s} ds + \frac{\partial w}{\partial t} dt$$

or

$$\frac{dw}{dt} = \frac{\partial w}{\partial s}\left(\frac{ds}{dt}\right) + \frac{\partial w}{\partial t} \tag{10}$$

where (ds/dt) describes the "characteristic" direction in the (s,t) plane along which Eq. (10) gives the rate of change of the parameter w. After the appropriate algebra and the introduction of Eq. (1), the final system of equations becomes

$$\left.\begin{aligned}
\frac{\partial Q}{\partial t} + (q+A)\frac{\partial Q}{\partial s} &= -\frac{\gamma-1}{2} A \left(S - \frac{2}{\gamma-1}\right)\left[\frac{\partial}{\partial s}\left(q - \frac{2}{\gamma-1} A\right)\right] \\
&\quad - \frac{\gamma-1}{2} q A S \left[\frac{\partial \theta}{\partial n} + \cos\theta \frac{\partial \phi}{\partial m}\right] \\
\frac{\partial R}{\partial t} + (q-A)\frac{\partial R}{\partial s} &= +\frac{\gamma-1}{2} A \left(S - \frac{2}{\gamma-1}\right)\left[\frac{\partial}{\partial s}\left(q + \frac{2}{\gamma-1} A\right)\right] \\
&\quad + \frac{\gamma-1}{2} q A S \left[\frac{\partial \theta}{\partial n} + \cos\theta \frac{\partial \phi}{\partial m}\right]
\end{aligned}\right\} \tag{11}$$

$$\left.\begin{aligned} \frac{\partial S}{\partial t} + q\,\frac{\partial S}{\partial s} &= 0 \\ \frac{\partial \theta}{\partial t} + q\,\frac{\partial \theta}{\partial s} &= -\frac{A^2}{\gamma q}\,\frac{\partial\, \ell n\, P}{\partial n} \\ \frac{\partial \phi}{\partial t} + q\,\frac{\partial \phi}{\partial s} &= -\frac{A^2}{\gamma\, q \cos\theta}\,\frac{\partial\, \ell n\, P}{\partial m} \end{aligned}\right\}$$

The characteristic directions in the space-time plane are clearly q+A, q-A, and q.

C. SOLUTION METHOD

Equation (11) may be expressed as

$$\frac{\partial \bar{w}}{\partial t} + [\lambda]\,\frac{\partial \bar{w}}{\partial s} = \bar{Z} \qquad (12)$$

or along the λ directions as

$$\frac{d\bar{w}}{dt} = \bar{Z} \qquad (13)$$

where the vectors are defined as

$$\bar{w} = \begin{Bmatrix} Q \\ R \\ S \\ \theta \\ \phi \end{Bmatrix} \qquad [\lambda] = \begin{bmatrix} q+A & & & & \\ & q-A & & & \\ & & q & & \\ & & & q & \\ & & & & q \end{bmatrix}$$

$\bar{Z}$ = right hand side of Eq. (11)

A trajectory in the space-time domain is illustrated in Fig. 1,

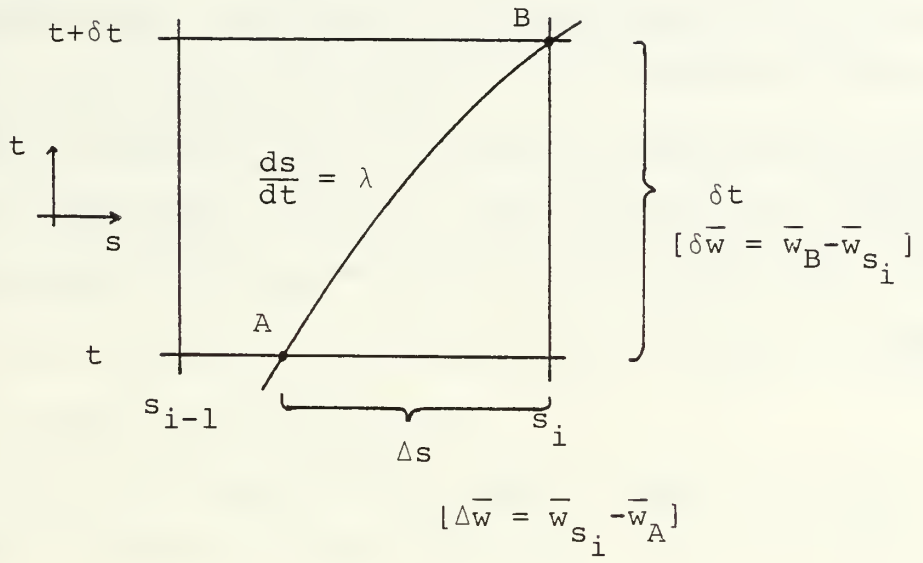


Figure 1. Solution Procedure in the Space-Time Domain

which shows an infinitesimal interval of space between two "nodes" of the computational mesh.

For the change in $\bar{w}$ from A to B along the trajectory with slope λ,

$$\int_{t}^{t+\delta t} \bar{Z}\, dt = \bar{w}_B - \bar{w}_A$$

$$= [\bar{w}_B - \bar{w}_{s_i}] + [\bar{w}_{s_i} - \bar{w}_A]$$

$$= \delta\bar{w} + \Delta\bar{w}$$

where $\delta\bar{w}$ denotes the change due to time at a fixed location and $\Delta\bar{w}$ denotes the change due to displacement at a fixed time, for the characteristic trajectory. The essence of the solution procedure is to calculate the change in the variables at each spatial node during the differential time interval, so that the step to the next time level can be made. In other words, $\delta\bar{w}$ must be calculated at each node using

$$\delta\bar{w} = -\Delta\bar{w} + \int_{t}^{t+\delta t} \bar{Z}\, dt \tag{14}$$

Since all the information at time level t is known, the position of A can be determined by an iterative procedure based on λ and $\Delta\bar{w}$ can be calculated by interpolation. The line integral can be evaluated by transforming the integral to a purely spatial integral using $\lambda = ds/dt$. Thus

$$\int_{t}^{t+\delta t} \bar{Z}\, dt = \int_{A}^{B} \frac{\bar{Z}}{\lambda}\, ds = \int_{s_i+\Delta s}^{s_i} \frac{\bar{Z}}{\lambda}\, ds$$

and the integral can be evaluated by any one of several numerical methods. Thus, the variables at each node can be updated in time using Eq. (14) and the process repeated.

D. DISCONTINUITIES IN THE FLOW

There are several types of discontinuities that must be considered. They are gradient discontinuities, contact discontinuities and shock waves.

Gradient discontinuities are characteristic of the head and tail of rarefraction waves, the collision of two shocks and the interaction of a shock and a contact discontinuity. Across such a gradient, the derivatives of velocity, pressure and sound speed are discontinuous.

A contact discontinuity is caused by the interaction of two shocks of opposite family and when a shock overtakes a shock of the same family. Entropy and sound speed are discontinuous at a contact discontinuity.

Across shock waves, velocity, pressure, density and entropy are discontinuous [Ref. 7].

Since Eq. (11) is not valid across discontinuities, additional logic is required in the numerical procedure to model flows which contain discontinuities. The method presented in Reference 6 for making a correction in the case of shock waves will give good accuracy in problems where the solution is converging to a steady state condition but will not give accurate results during the transient portion of such problems or for problems with only unsteady solutions. In the case of applications to the wave rotor, it would certainly be necessary to know the locations of the contact surfaces between the

hot and cool gases as well as the locations of the shock and expansion waves.

The methods used in the present work to correct for discontinuities are those of Moretti [Ref. 7]. The present discussion will be limited to the treatment of shock waves. The method makes use of the analytical relationship between the change in the Riemann variables across a shock and the incoming Mach number relative to the shock wave (W) which is illustrated in Figs. 2 and 3. The relation is used to determine the shock speed and to transform the problem to a steady case which can be handled using normal shock relations. For the situation depicted in Fig. 2 of a shock propagating to the right with velocity V_s into air with velocity q_B, and with the high pressure side to the left, where A denotes the left side of the shock and B the right,

$$W = -\frac{u_B}{A_B} = \frac{-(q_B - V_s)}{A_B} \tag{15}$$

If the pressure and density are non-dimensionalized by the values on the low pressure side of the shock, the change in the extended Riemann variable Q is given by

$$\frac{Q_A - Q_B}{A_B} = \frac{2(W^2-1)}{(\gamma+1)W} + \frac{2}{\gamma-1}\left[\frac{A_A}{A_B} - 1\right]$$

$$- \frac{A_A}{A_B}\left[\frac{1}{\gamma(\gamma-1)}\right] \ln\left\{\left[\frac{2\gamma}{\gamma+1} W^2 - \left(\frac{\gamma-1}{\gamma+1}\right)\right]\left[\frac{(\gamma-1)W^2+2}{(\gamma+1)W^2}\right]^{\gamma}\right\} \tag{16}$$

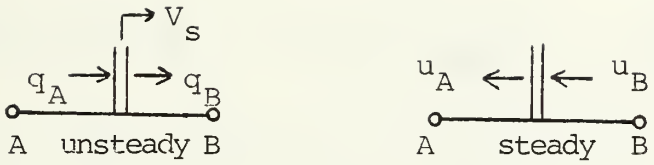


Figure 2. Shock Wave with High Pressure to the Left

Figure 3. Shock Wave with High Pressure to the Right

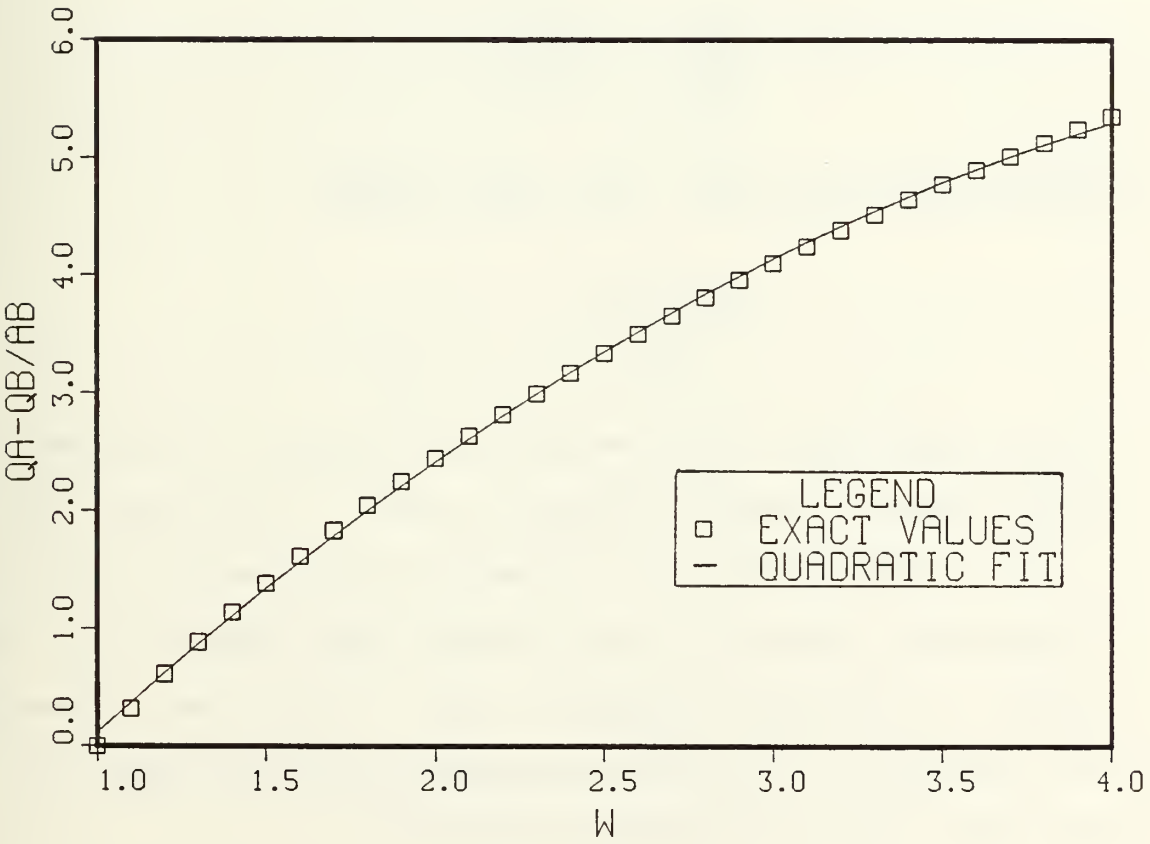


Figure 4. $(Q_A-Q_B)/A_B$ vs W

where

$$\frac{A_A}{A_B} = \frac{1}{(\gamma+1)W}\left\{2(\gamma-1)\left[1 + \frac{\gamma-1}{2} W^2\right]\left[\frac{2\gamma}{\gamma-1} W^2 - 1\right]\right\}^{1/2}$$

This exact relationship can be approximated by the polynomial

$$\frac{Q_A - Q_B}{A_B} = -2.7574 + 3.1573W - 0.2863W^2 \qquad (17)$$

as illustrated in Fig. 4 over a shock strength range of 1.0 to 4.0. It should be noted that if the high pressure side were to the right, as in Fig. 3, Eq. (15) would become

$$W = \frac{u_A}{A_A} = \frac{q_A - V_s}{A_A}$$

and the left side of Eq. (16) would become

$$\frac{R_A - R_B}{A_A}$$

The procedure then, is to measure the change in Q across an interval where a shock is known to exist. This value, call it ΔQ_m, is used in Eq. (17) to get an approximation for the corresponding value of W and then the exact value of ΔQ, call it ΔQ_E, is calculated using Eq. (16). If the exact value of ΔQ is not equal to the value measured across the shock, a new value, ΔQ, is calculated according to

$$\Delta Q_{i+1} = \Delta Q_i + (\Delta Q_m - \Delta Q_E)$$

and entered into Eq. (17) to obtain another value of W. When the exact value of ΔQ calculated from Eq. (16) equals the measured value of ΔQ, W is known to be correct and the normal shock relations can be used to calculate the values at node A. Also, V_s can be calculated from Eq. (15) and used to track the shock during the next time interval.

III. DESCRIPTION OF FORTRAN PROGRAM "EULER-1"

A. MACHINE AND LANGUAGE

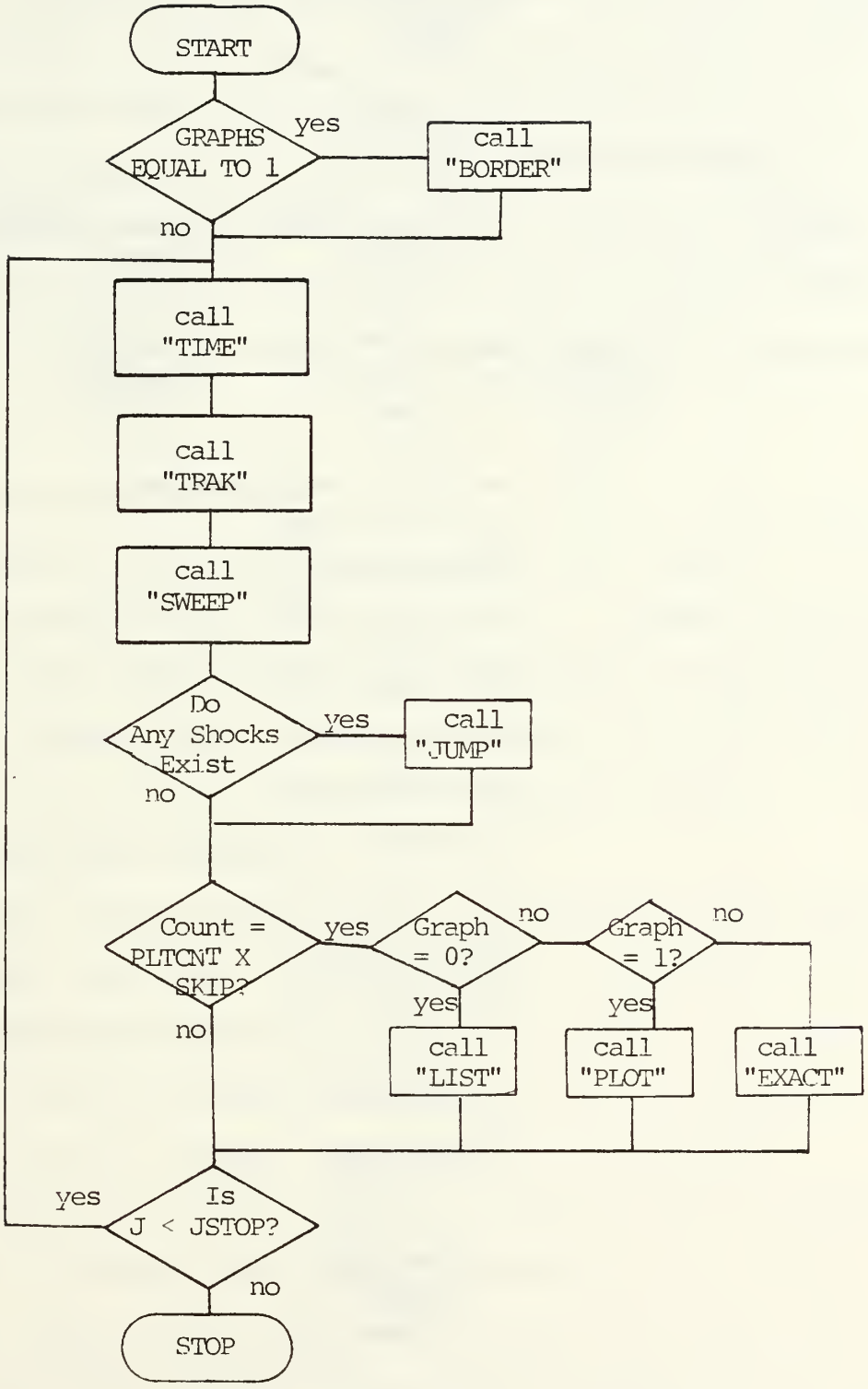
The EULER-1 program is written in VS FORTRAN and runs on an IBM 3033, System 370 computer, however the code is simple and small enough to enter and run on a mini or micro computer in a Basic language if one is willing to accept significantly longer run times. Table 1 at the end of this section is a summary of the editable parameters and their effect on the program.

B. CONVENTIONS AND BASIC STRUCTURE

EULER-1 is a first-order one-dimensional code using an evenly spaced numerical grid. All values are double precision except those used in the graphics routines which are rounded to single precision.

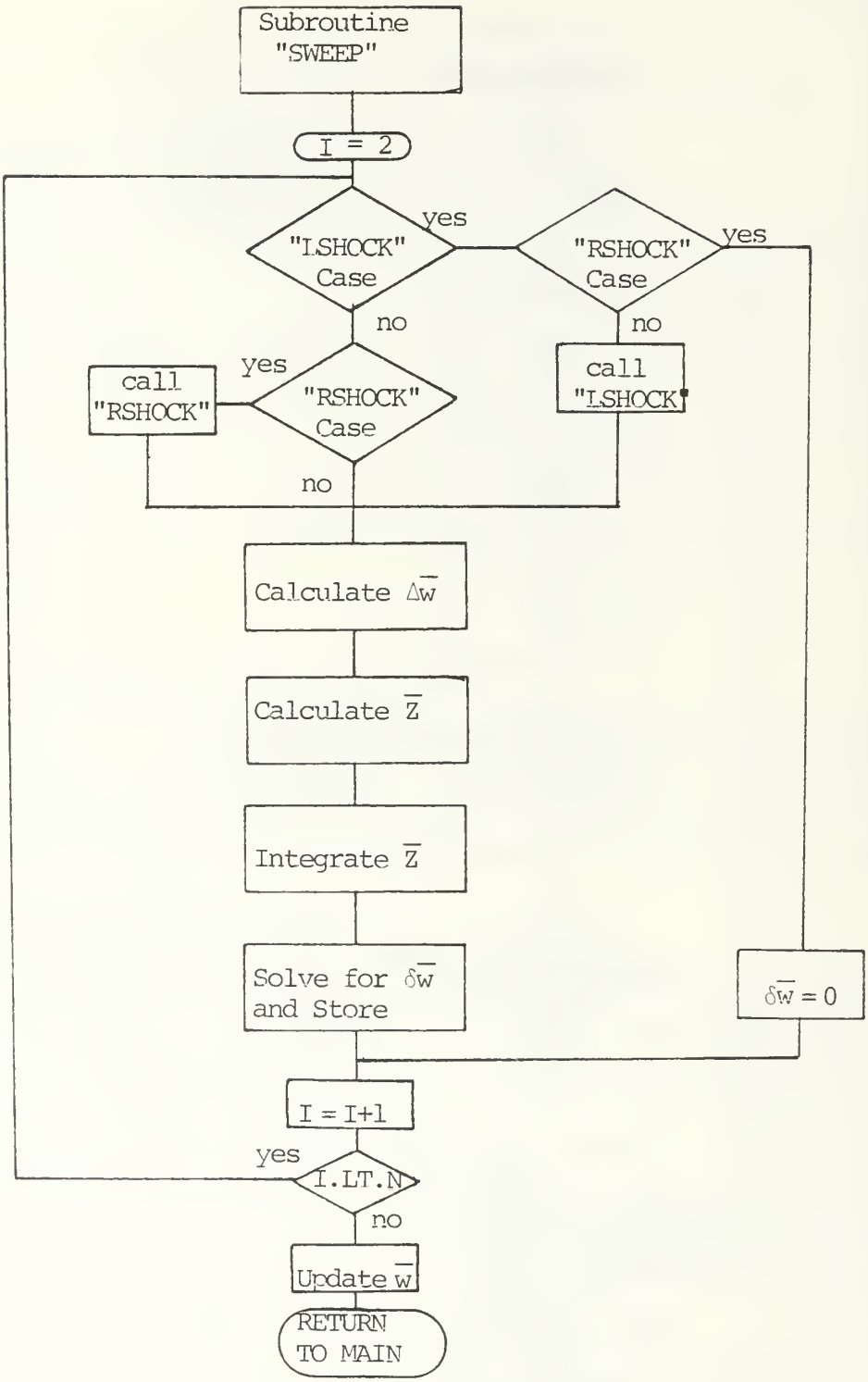
Each subroutine has its own variables space. In other words, variables are not shared in common throughout the program but must be passed to the subroutine being called by the calling routine. In all cases, however, the variables have the same name in both the called and the calling routine so there should be no confusion. This was done so that arrays could be dimensioned at execution time.

The program, depicted in Fig. 5, is structured around a main routine which serves as a user input area, sets up the problem and calls five subroutines to solve the problem and



a) Main Program

Figure 5. EULER-1 Flowchart



b) "SWEEP" Subroutine

Figure 5. (Continued)

output the solution. The five subroutines are "TIME," "TRAK," "SWEEP," "JUMP" and then one of several output routines depending on user desires. Output options include two types of graphical displays, "PLOT" and "EXACT," and one tabular listing routine, "LIST." When "PLOT" is selected, "BORDER" is automatically called to set up the plotting area. There are two other subroutines, "RSHOCK" and "LSHOCK," which are called by "SWEEP" as needed.

C. SUBROUTINE DESCRIPTIONS

In general, each subroutine begins with a heading followed by variable definitions where appropriate, variable declarations and array dimensioning. Most variables are defined in the "MAIN" routine and only those variables which were not are defined in the subroutines where they are used.

1. The "MAIN" Routine

This routine forms the main structure of the program and includes a heading, an extensive list of variable definitions, a user input area, the necessary statements to make the initial value assignments to variables, and the call statements for the various subroutines.

The input area is those lines of the program (145-175) where the user edits the program to establish the initial conditions, mesh size, termination criteria and output options.

The initial conditions which may be modified are the pressure, temperature and density ratios across the diaphragm, the initial velocity of the fluid on each side of the

diaphragm, and the value of gamma (which must be the same for both sides).

All velocities are non-dimensionalized by the initial sound speed on the low pressure side of the diaphragm and pressures and densities are non-dimensionalized by their initial values on the low pressure side of the diaphragm.

The mesh size may be set at any odd number and the arrays in the user input area must be dimensioned as such.

The criteria for program termination is the number of time steps computed, which the user selects.

The output options are determined by the value of the variable GRAPHS. A value of zero results in a call to "LIST" which writes to file 9 on the user's permanent disk, a tabular listing of the variable arrays and discontinuity locations. A value of 1 results in a call to "BORDER" and "PLOT" which create a plot of the pressure, density, velocity and entropy distributions as in Fig. 7. A value of 2 results in a call to "EXACT" which creates a plot of the density distribution compared to the exact solution as in Fig. 8. The exact values to be plotted must be entered in the user input area, they are not calculated by the program. A more detailed description of the various outputs is found under the appropriate subroutine description. The frequency with which output is created is controlled by the variable SKIP which is the number of time steps between calls to output routines.

2. The "TIME" Routine

The maximum allowable time step is governed by the CFL condition. Simply stated, this means that the time step must be small enough so that the characteristic trajectories remain within one spatial interval during the time interval. The minimum time step is calculated by computing

$$\text{Delt} = H/\text{ABS}[Q+A]$$

at every node, where H is the spatial interval, and selecting the minimum value of Delt.

3. The "TRAK" Routine

Shock locations at the unknown time level are determined by computing the shock speed at the known time level, as outlined in Section 2, and multiplying by the time interval computed by "TIME." The time step is reduced, if necessary, to limit shock travel to one spatial interval. The node immediately to the right of the shock (upstream) is flagged for later use by "SWEEP" and "JUMP."

In calculating the shock speed, the assumption is made that the conditions immediately adjacent to the shock are the same as the conditions at the nodes to the left and right of the shock.

4. The "SWEEP" Routine

The "SWEEP" routine makes the necessary calculations to solve Eq. (14). This involves interpolation at the known

time level for the values of $\Delta\bar{w}$, calculation of $\bar{z}$, and integration of $\bar{z}$ for each node.

To calculate $\Delta\bar{w}$, the assumption is made that the characteristic trajectories are straight lines. An initial guess of the characteristic slope (λ) is made and by forcing the characteristic to pass through the node at point B in Fig. 1, $\Delta s = \lambda \times$ delt. The values of q and A at point A are found by interpolation and the slope of the characteristic is then calculated using the values of q and A at point A. This slope is compared with the initial guess and if the two are not in agreement, the procedure is iterated until they converge. Once Δs is accurately determined, the values of $\Delta\bar{w}$ can be calculated by interpolation. Linear interpolation is used in the present version of EULER-1. This procedure is carried out for each characteristic at each node.

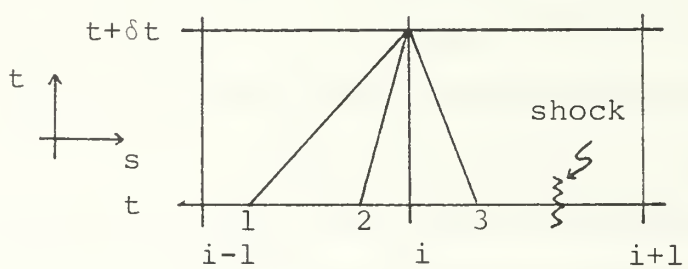


Figure 6. "RSHOCK" Case

A deviation from the above procedure is necessary in the case where a shock exists in the spatial interval to the left or right of the current node. When a shock exists to

the right, and the flow is subsonic, as in Fig. 6, care must be taken not to interpolate for point 3 based on the slope between points I+1 and I, which would not be accurate due to the discontinuity. In such a case "RSHOCK" is called and the interpolation is based on the slope between point I and I-1. Similarly, when the shock is to the left of the current node, "LSHOCK" is called and the interpolation is based on the slope between I+1 and I.

The calculation of $\bar{z}$ includes the calculation of spatial derivatives of q and A for characteristics 1 and 3 of Fig. 6. In general, the derivatives associated with characteristic 3 are forward differenced and those associated with characteristic 1 are backward differenced in keeping with the principle of domain of dependence. If the flow is supersonic or if a shock exists to the right of the current node, all derivatives are backward differenced. If a shock exists to the left of the current node, all derivatives are forward differenced. Once the derivatives are known, $\bar{z}$ is calculated from Eq. (11) using the average value of A between points 1 and I or 3 and I as appropriate. Since the derivatives are linear, this results in an average value of $\bar{z}$ over the same interval.

The integration of $\bar{z}$ is transformed from a time to a spatial integration as described in Section II and the trapezoidal rule is used to carry out the integration. In EULER-1, this has been done in one step using the average value of $\bar{z}$ described above.

Equation (14) is then solved by addition of $\Delta\bar{w}$ and the results of the integration, and the resulting $\delta\bar{w}$ is stored until all nodes are calculated. After all nodes have been calculated, the variable arrays ($\bar{w}$) are updated for the next time interval.

5. The "LSHOCK" Routine

As discussed above, the "LSHOCK" routine modifies the basic EULER-1 procedure at an interior node when a shock exists to the left of the node. Interpolation of quantities to the left of the node are computed based on the derivatives of the quantities to the right of the node. Although this assumes that the derivatives do not change between adjacent spatial intervals, it is necessary to avoid taking derivatives across discontinuities in the flow.

6. The "RSHOCK" Routine

Similar to "LSHOCK," "RSHOCK" bases interpolations of quantities to the right of the node on the derivatives of the quantities to the left of the node when a shock exists to the right of the node.

7. The "JUMP" Routine

The "JUMP" routine is used to calculate the conditions downstream of a stationary shock as described in Section II. If a shock is known to have crossed a spatial node during a time interval, which is known once "TRAK" has been called for that time interval, the entire "SWEEP" sequence is skipped for the node which was crossed by the shock and the conditions at

that node are determined using the normal shock relations. Note that the node in question is the node downstream of the stationary shock. As in "TRACK," it is assumed that the conditions at the nodes upstream and downstream of the shock are the same as the conditions immediately adjacent to the shock.

8. The Output Routines

There are four subroutines which produce various types of output as the user desires. "BORDER" and "PLOT" produce a graphical presentation of the pressure, density, velocity and entropy distributions at selected time levels as seen for example, in Fig. 7. "EXACT" produces a plot of the density distribution at one selected time interval and compares it with the exact solution as shown in Fig. 8. At selected time levels, "LIST" produces a tabular listing of the Riemann variables, modified entropy, pressure, density and velocity distributions, elapsed time, shock speed and discontinuity locations. The listing is written to the user's permanent storage disk.

When the value of GRAPHS is set equal to one in the "MAIN" routine, "BORDER" is called once to set up the plot axis, labels, and headings. "PLOT" is called every SKIP time steps to draw the four distribution curves.

When the value of GRAPHS is set equal to two, "EXACT" is called every SKIP time steps to plot the density distribution compared with the exact solution computed at six points of interest. The points are the two end points, which

are simply the initial conditions, the head and tail of the rarefraction wave, the point just left of the contact surface and the point just left of the shock. The spatial locations of these points are computed by "EXACT" based on the elapsed time and the known values of the wave velocities entered in the "MAIN" routine. The exact values of the densities at these points must also be entered in the "MAIN" routine.

When the value of GRAPHS is set equal to zero, the tabular listing as described above is sent to the user's disk. No graphical output is created.

D. CAPABILITIES AND LIMITATIONS

The present version of EULER-1 is set up to solve a shock tube problem with a single centered diaphragm. Boundary conditions for the ends of the tube have not been incorporated so the problem must be stopped before the waves reach the end of the tube.

The left side of the diaphragm is the high pressure side. The program will not run with the right side as the high pressure side without changes to some of the shock correction logic.

The user may select any odd number of grid points limited only by the amount of memory available.

The program tracks shock waves and makes shock jump calculations at the appropriate locations but the program can also run without the shock tracking feature and jump calculations if so desired with some smearing of the shock discontinuity and complete loss of entropy change across the shock.

TABLE 1

LIST OF EDITABLE PARAMETERS

Parameter	Function
N	Number of grid points
M	Controls which discontinuities are tracked 1 = Shocks only 2 = Shocks and contact surfaces
GRAPHS	Controls the form of the output 0 = Tabular listing 1 = Pressure, density, velocity, entropy plot 2 = Exact solution comparison for density
SKIP	Number of time steps between output calls
JSTOP	Number of time steps calculated
TRI	Initial temperature ratio
PRI	Initial pressure ratio
DRI	Initial density ratio
QLI	Initial velocity left of the diaphragm
QRI	Initial velocity right of the diaphragm
G	Ratio of specific heats

IV. TEST RESULTS

EULER-1 was tested on the Riemann shock tube problem. The initial conditions and the courseness of the computational mesh were varied. Additionally, one run was made without the shock tracking and correction features.

Figures 7 and 8 show the results for initial pressure and density ratios of 5, uniform temperature and with the air initially at rest, using a grid of 101 points. The exact, analytically predicted conditions are also shown in Fig. 8. It can be seen that all wave velocities are correctly computed and the shock is defined within one interval. The rarefraction waves are slightly smeared and the contact surface is greatly smeared. A slight transient instability to the right of the diaphragm is noticeable, particularly in the plots of pressure and velocity.

Figures 9 and 10 show results for initial pressure and density ratios of 1.3. The shock is still well defined in the correct interval and the contact surface is not as badly smeared as in Fig. 8. There is a loss of accuracy however, with respect to the tail of the rarefraction wave.

Figure 11 shows the results of the original problem with a grid of 51 points. As might be expected with a coarser mesh, the discontinuities are less sharply defined although it is clear that the wave velocities are accurately computed.

Figure 12 shows the results for the original problem with a grid of 901 points. It can be seen that the EULER-1 solution closely approximates the exact solution.

Figures 13 and 14 show the result of running the original problem without the shock tracking and correcting features. Note that the shock position is incorrectly computed and it is smeared slightly. Also note in Fig. 13, the lack of any change in modified entropy across the shock.

Run time for the 101 point mesh is 0.00049 seconds per node per time step. The results for Fig. 7 took 50 time steps for a total time of 2.46 seconds. For the 901 point mesh, the run time decreased to 0.00023 seconds per node per time step due to the less frequent use of the shock tracking and correction routines per node calculated. The results for Fig. 12 took 470 time steps for a total time of 71.85 seconds.

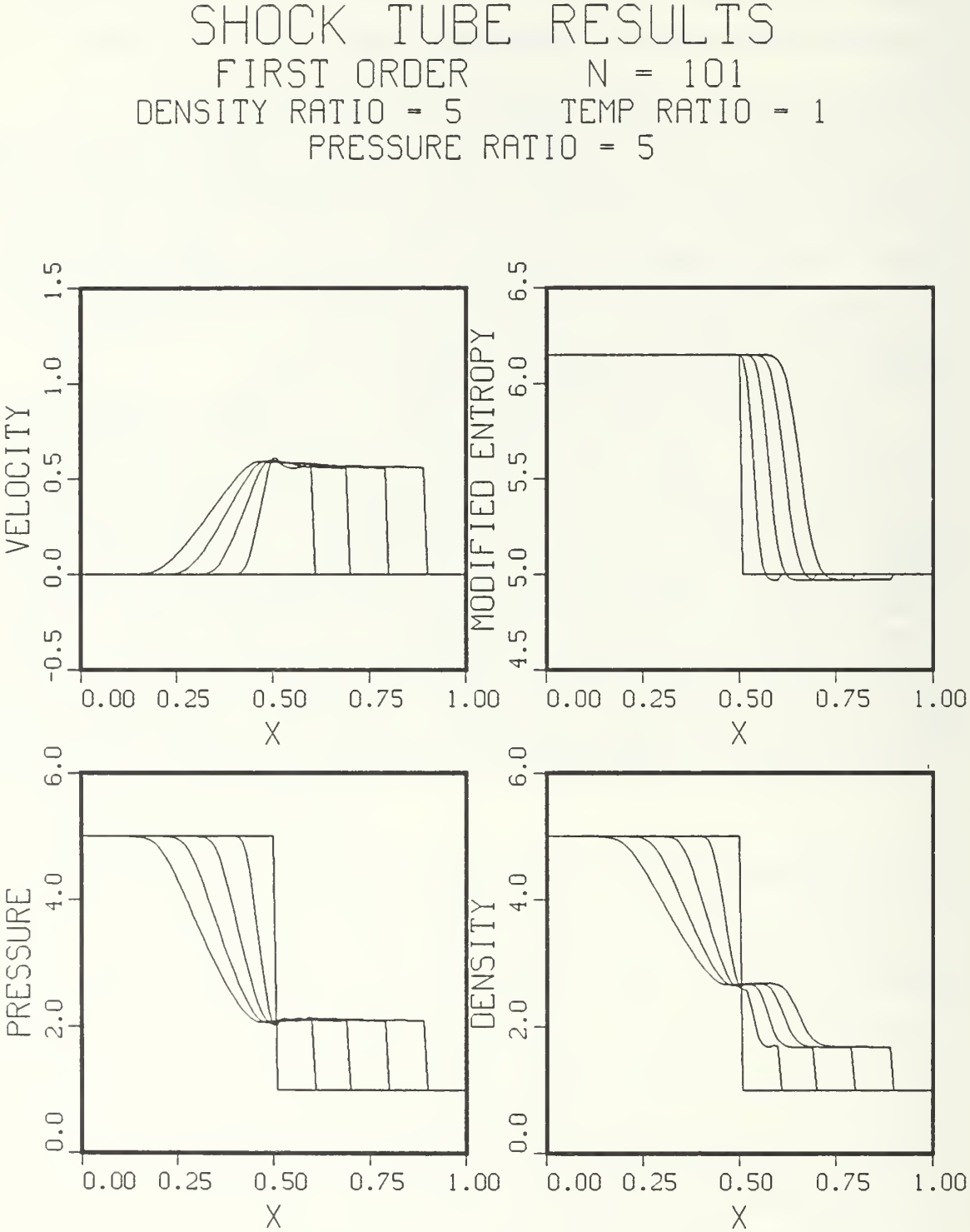


Figure 7. EULER-1 Results for Pressure Ratio 5 with Shock Tracking

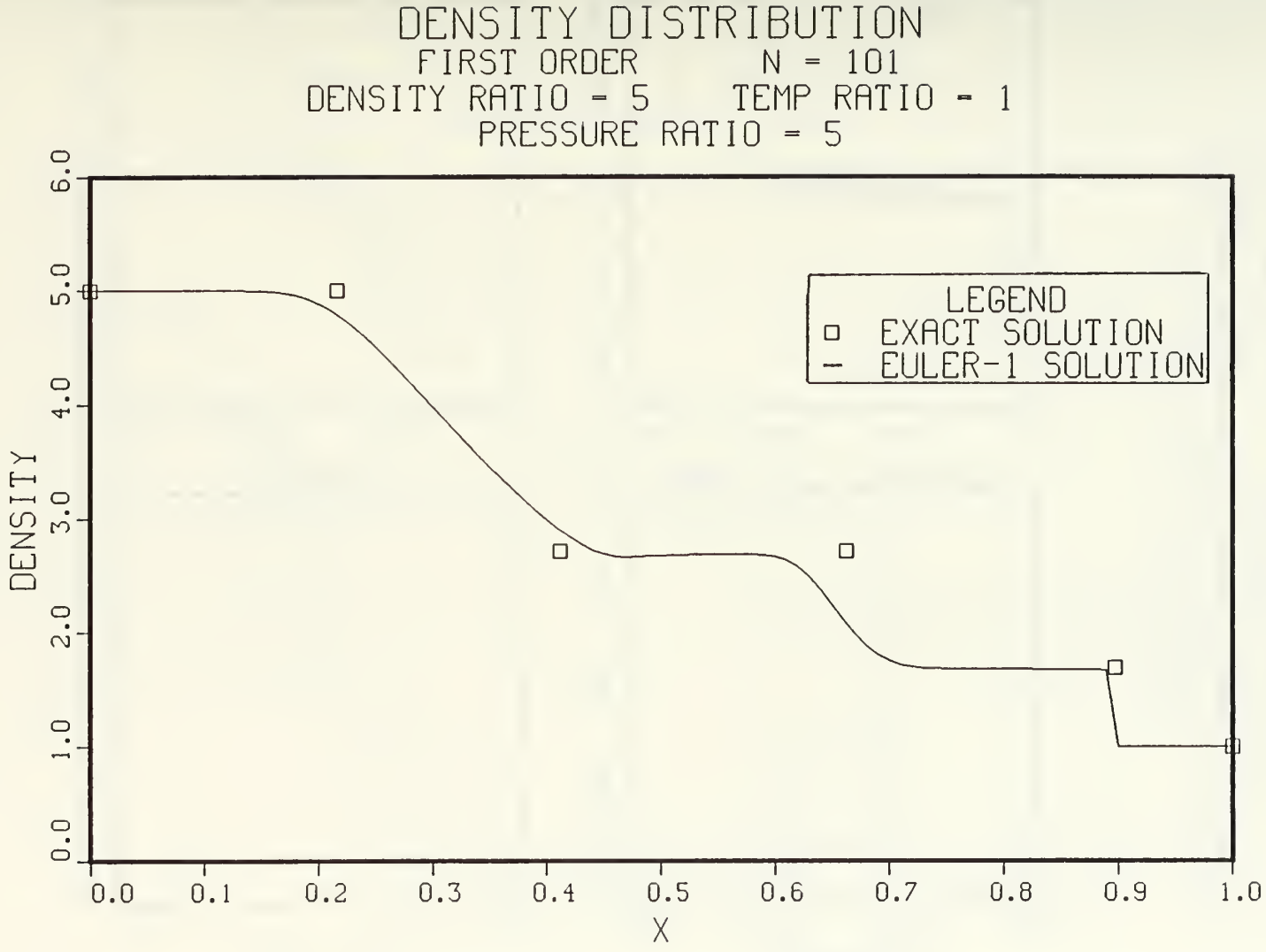


Figure 8. Exact Solution Comparison for Pressure Ratio 5, with Shock Tracking

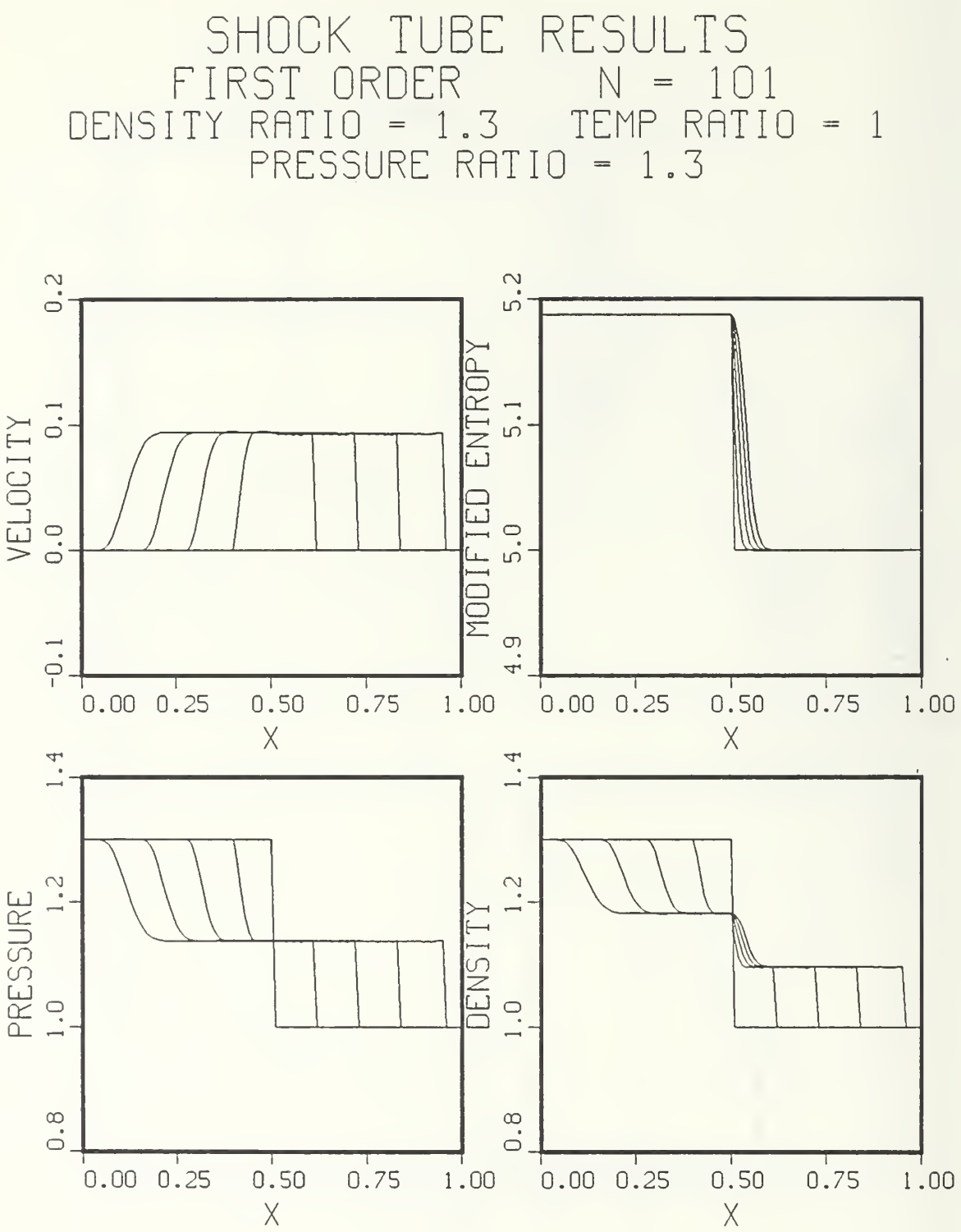


Figure 9. EULER-1 Results for Pressure Ratio 1.3 with Shock Tracking

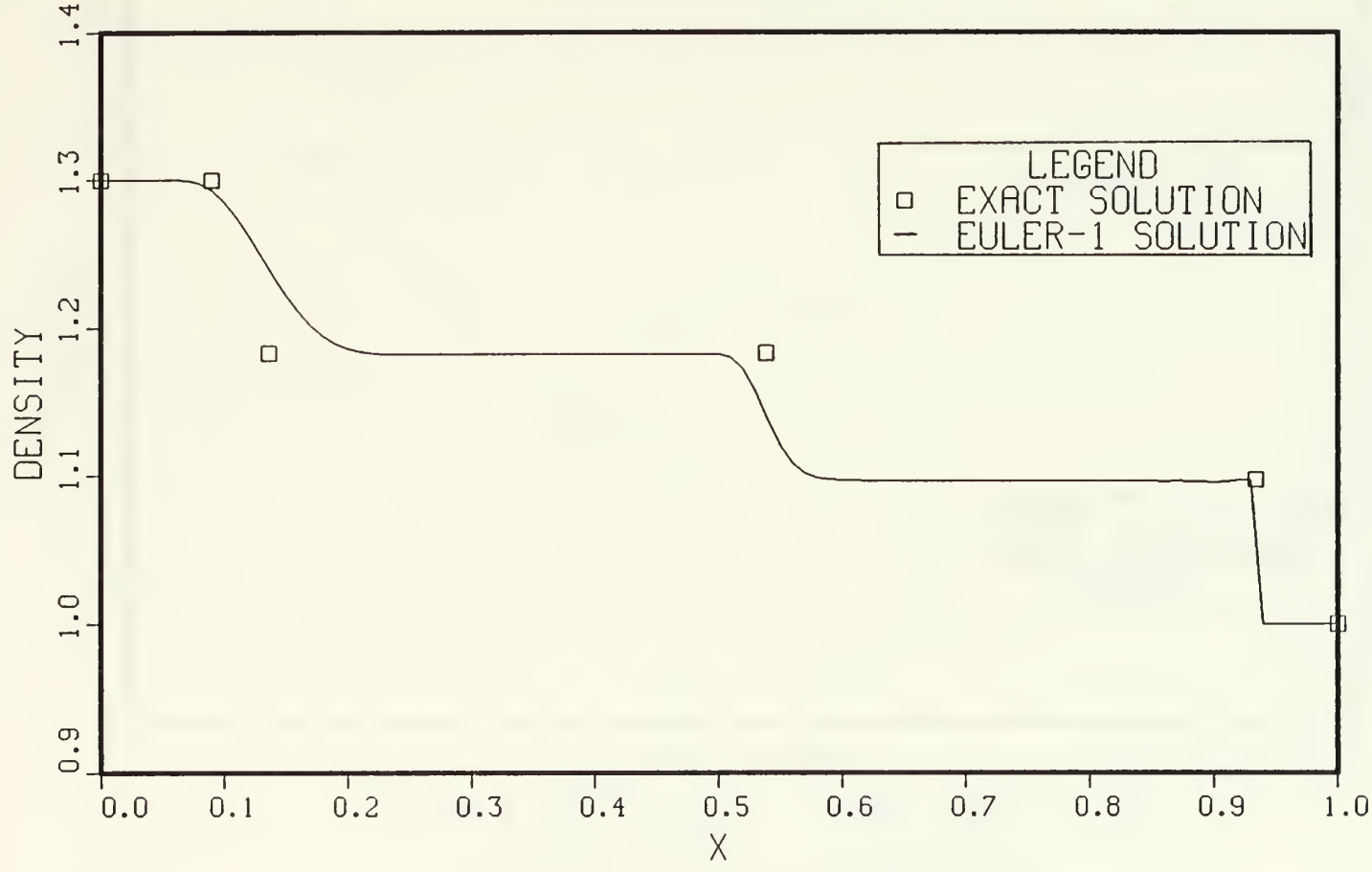


Figure 10. Exact Solution Comparison for Pressure Ratio 1.3, with Shock Tracking

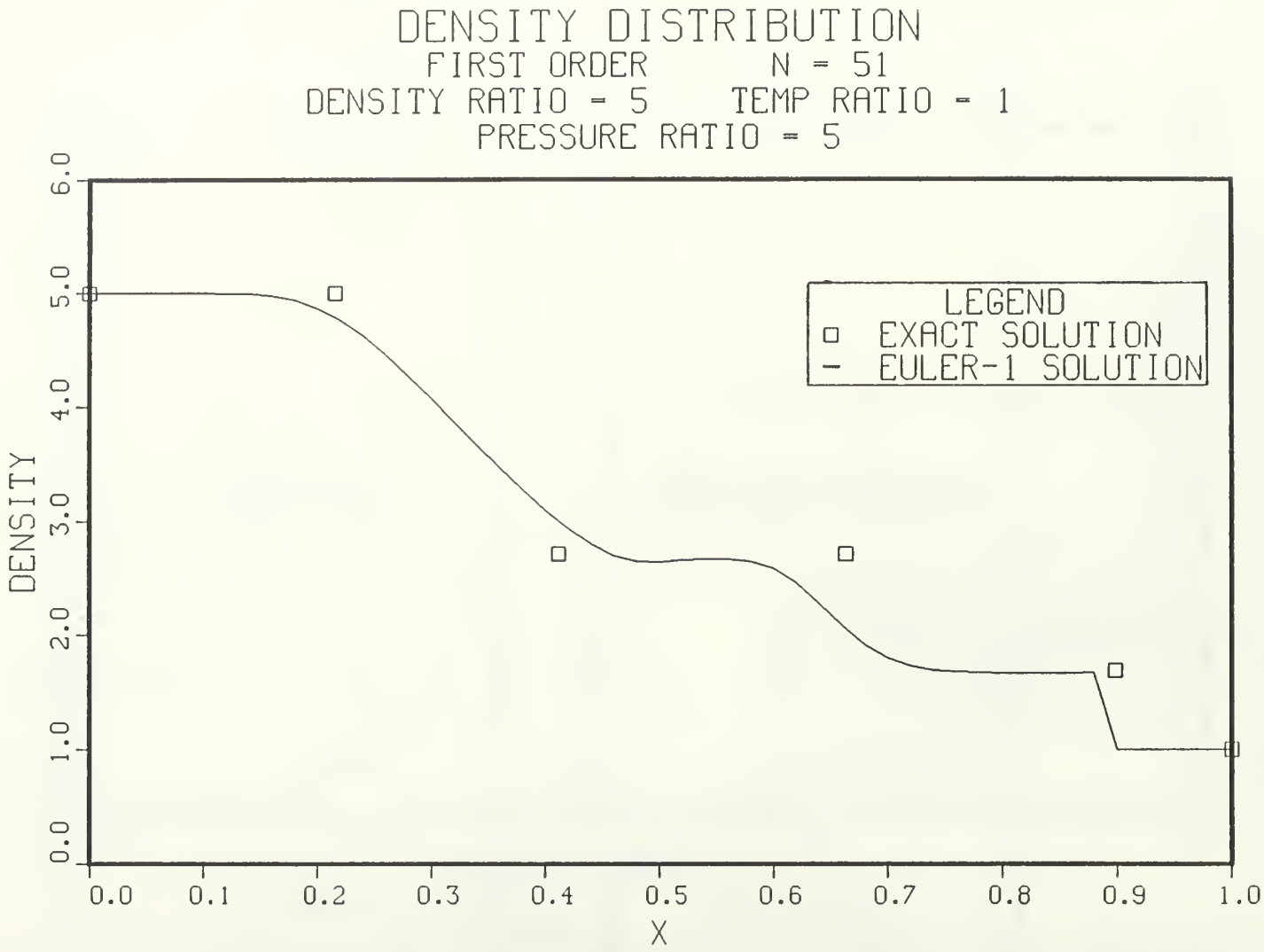


Figure 11. Course Mesh Solution for Pressure Ratio 5

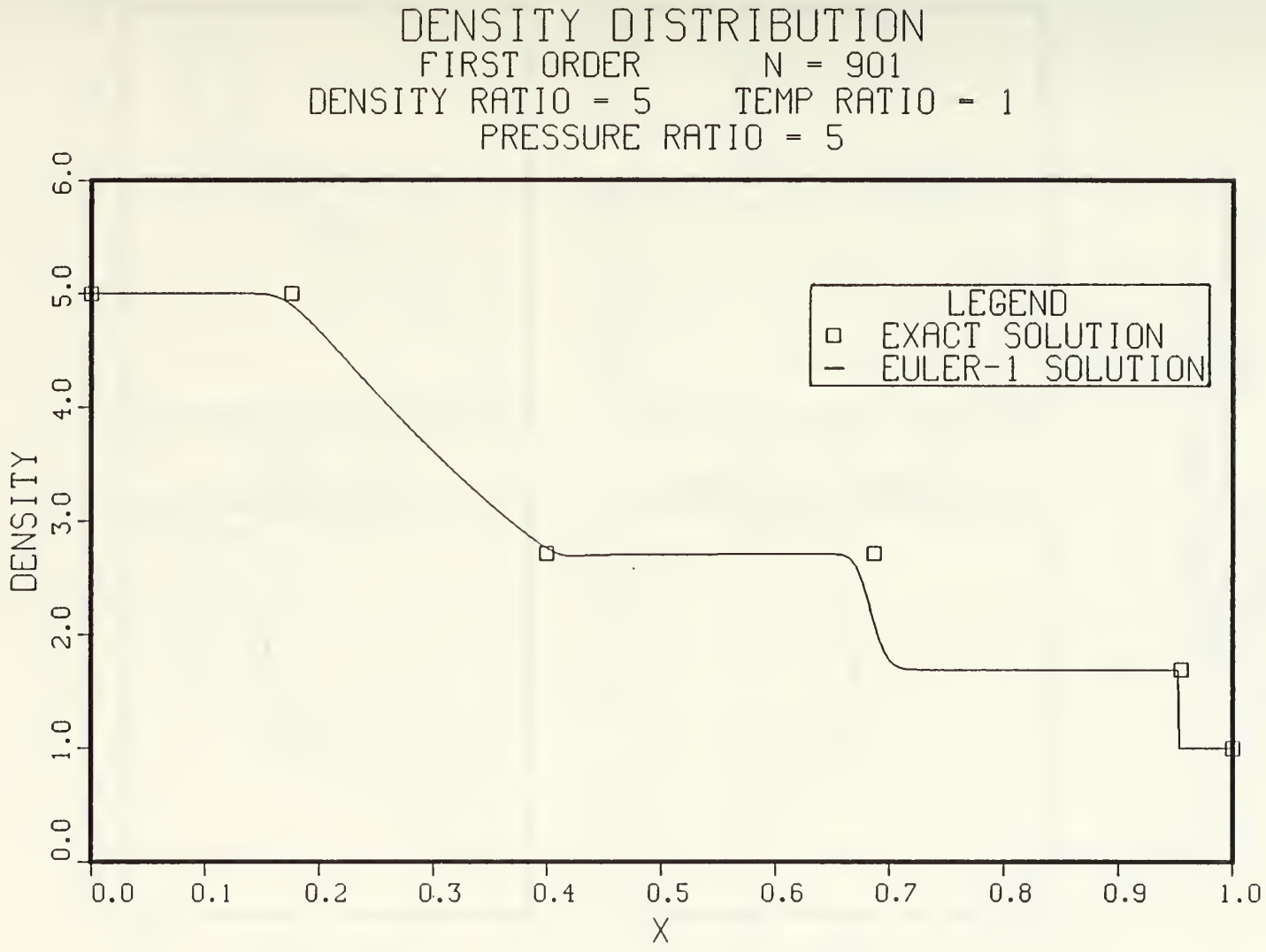


Figure 12. Fine Mesh Solution for Pressure Ratio 5

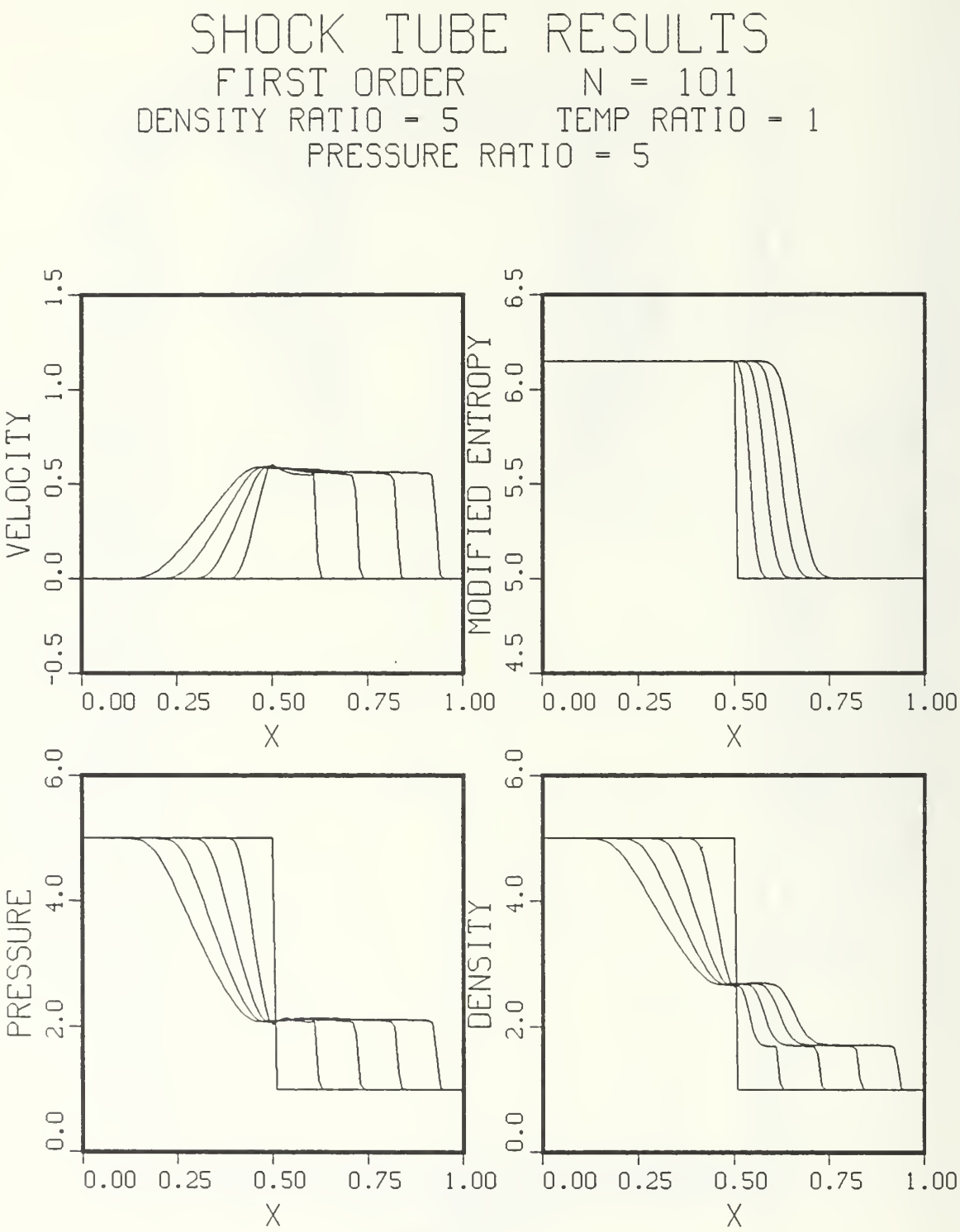


Figure 13. EULER-1 Results for Pressure Ratio 5 without Shock Tracking

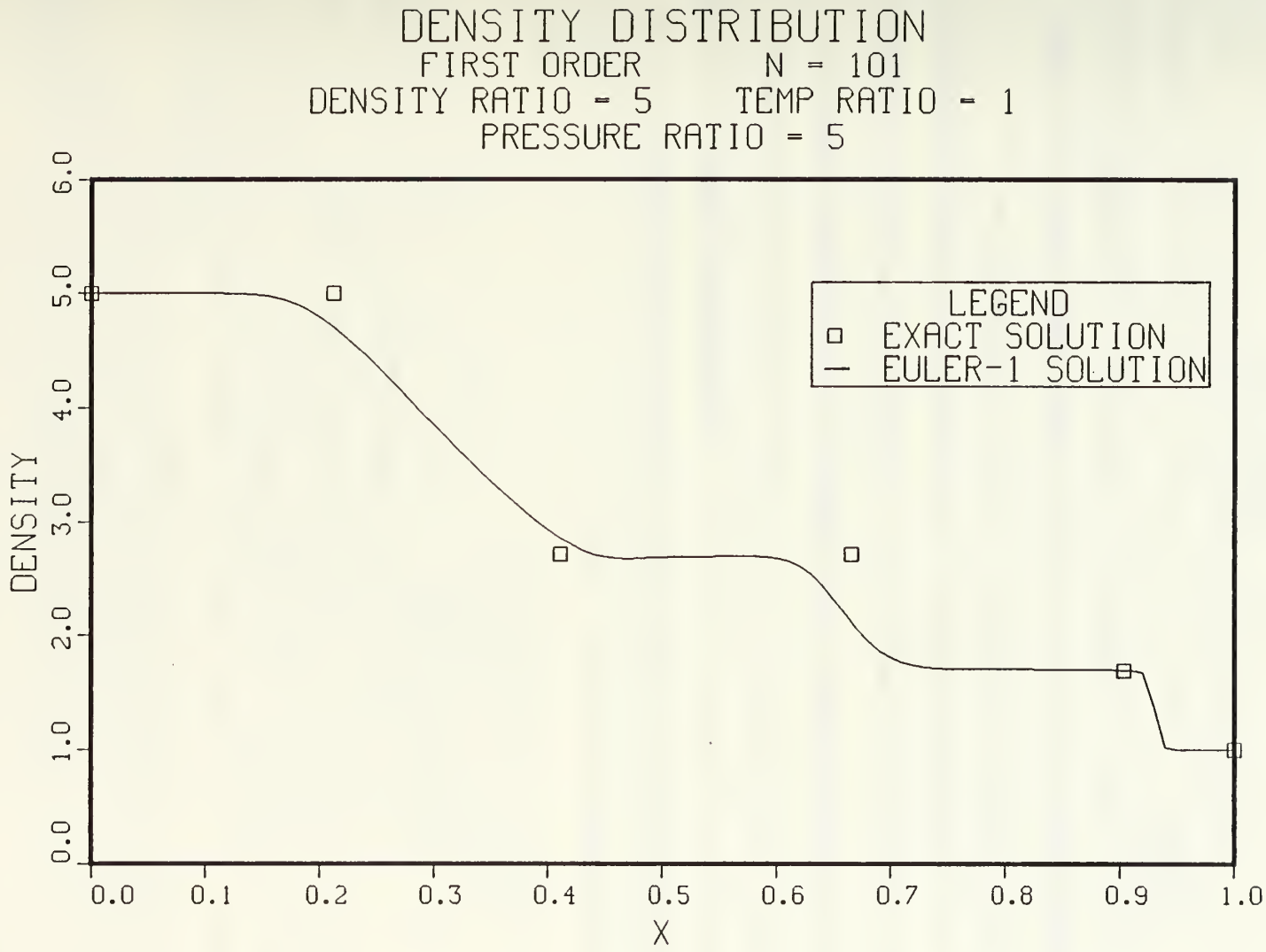


Figure 14. Exact Solution Comparison for Pressure Ratio 5, without Shock Tracking

V. DISCUSSION

A. RESULTS

The transient instability noted in the results was not investigated since the amplitude was small and the effect would be of no consequence in most applications.

At low pressure ratios, the velocity of propagation of the tail of the rarefraction wave approached sound speed and, in these cases, the EULER-1 code did not accurately compute its location. The reason for this was not investigated here; however, due to the low pressure ratios at which wave rotors can operate, this is of interest and should be investigated in future work.

B. SPECIAL CONSIDERATIONS

1. Modified Entropy

In the QAZ1D method, and subsequently, in the EULER-1 code, the variable S, which here has been referred to as the "modified entropy" and which is referred to in Reference 6 as simply the "entropy," does not behave thermodynamically as one would expect entropy to behave. As a fluid crosses a shock wave, the modified entropy of the fluid decreases. This is the result of the definition of Eq. (3), repeated here, that

$$dS = -\frac{1}{\gamma}\frac{dQ_R}{T}$$

The use of the word "entropy" and symbol "S" for this variable is confusing since the sign of the change in "S" is in direct conflict with well-established conventions in thermodynamics. For example, Corollary 6 to the 2^{nd} Law of Thermodynamics [Ref. 8:p. 88] would have to be changed to read, "The 'modified entropy' of an isolated system decreases or in the limit remains constant." At a minimum perhaps, the symbol $\bar{S}$ to denote the "modified (negative and scaled) entropy" would be helpful. In practice of course, the change in the conventionally defined entropy can be recovered from the change in the modified entropy, simply by multiplying by $(-\gamma)$.

2. Moretti's Methods

Incorporating Moretti's methods of handling discontinuities [Ref. 7] into the EULER-1 code required special care.

There is a fundamental difference in procedure in that the EULER-1 code is based on the high pressure side being on the left whereas Moretti's formulations are all based on the high pressure side being on the right. Although essentially a matter of bookkeeping, it is an area of potential confusion.

Another difference is that the Riemann variables used by Morreti are not the Riemann variables used in the QAZ1D method. This can lead to difficulty. In fact, in the application of Moretti's shock tracking scheme to the EULER-1 code, the pressures and densities have to be non-dimensionalized

by their values on the low pressure side of the shock in order to obtain Eq. (14), and for the procedure to work.

C. SUITABILITY FOR WAVE ROTOR APPLICATIONS

Further work is necessary before the suitability of the QAZ1D method for wave rotor applications can be fully evaluated. The following areas need to be addressed.

1. Boundary Conditions

In the EULER-1 code, boundary conditions at the ends of the passage have not been incorporated. The code calculates the changes of the interior nodes only, skipping the two end nodes. In particular, solid wall boundary conditions and open-end boundary conditions must be incorporated. No particular problems are expected in the boundary conditions themselves. However, additional logic may be necessary in the handling of discontinuities, such as the reversal of conditions to the left and right of a discontinuity after reflection from a boundary. The additional logic, which may be considerable, is warranted by the simplicity of the QAZ1D method.

2. Contact Discontinuity

No special attention is given to the contact discontinuity in the EULER-1 code and hence the discontinuity is smeared. In a wave rotor application, this discontinuity will have to be sharp. Moretti's methods again appear to be applicable here and the additional logic needs only to be formulated and incorporated into the EULER-1 code.

3. Quasi-One-Dimensional Modeling

EULER-1 is a one-dimensional code. The solution of flows in passages of varying area may be desirable for some wave rotor applications. Such problems can be solved in a quasi-one-dimensional manner by the addition of the appropriate area variation term to the equations, and the need to solve the fully two-dimensional equations is avoided. The area variation must be incorporated into the EULER-1 code and the code tested on quasi-one-dimensional problems.

4. Other Potential Extensions

Another capability which can be incorporated is the ability to handle two gases with different specific heat ratios. Eventually, the code should be extended also to a second order, one-dimensional version and then to a first order, two-dimensional version.

VI. CONCLUSIONS

The development of the EULER equations in the natural streamline coordinates using extended Riemann variables was reviewed in detail. The QAZ1D solution method, with the addition of shock tracking features, was implemented in a first order, one-dimensional FORTRAN code with the intent of evaluating the method's suitability for wave rotor applications. The code (EULER-1) was tested on the shock tube problem with good results. The incorporation of boundary conditions, an improvement in the contact discontinuity definition and the addition of an area variation term for quasi-one-dimensional modeling are considered to be necessary before the QAZ1D method can be accepted as being suitable for wave rotor applications. The additional logic required for these extensions may be considerable but the development is warranted by the overall simplicity of the QAZ1D method, and its straightforward extension to viscous, multi-dimensional flow modeling.

APPENDIX A

DERIVATION OF THE GOVERNING EQUATIONS FOR THREE-DIMENSIONAL INVISCID FLOW (WITH AREA CHANGE)

The governing equations of a 3-D, inviscid, compressible, unsteady flow are derived here in a natural streamline coordinate system. The equations are then recast along characteristic trajectories in the (s,t) plane and expressed in extended Riemann variables, reducing the system of partial differential equations to a system of ordinary differential equations.

A. DEFINITION OF VARIABLES

A	Cross-sectional area of a differential stream tube
A	Speed of sound
C_p	Specific heat at constant pressure
C_v	Specific heat at constant volume
$d\vec{a}$	Normal vector for differential area of control surface
e	Specific internal energy
$\hat{i}_{s,n,m}$	Unit vectors in the s, n, and m directions
P	Static pressure
Q	Modified Riemann variable
Q_R	Reversible heat transferred
q	Velocity magnitude
$\bar{r}$	Position vector

R	Modified Riemann variable
R_G	Gas constant
S	Modified entropy
T	Static temperature
t	Time
u	Velocity relative to a standing shock wave
v	Specific volume
vol	Differential element of volume
$\bar{V}$	Velocity vector
W	Incoming Mach number relative to a standing shock wave
ρ	Density
γ	Ratio of specific heats
θ,ϕ	Flow angles with respect to reference coordinate planes.
∇	Vector operator

B. THE COORDINATE SYSTEM

The natural streamline coordinate system (s,n,m) is shown with respect to a fixed rectangular cartesian system (x,y,z) in Fig. A1. The system is a curvilinearly translating, right hand orthogonal system which translates with a fluid particle along a streamline such that the 's' coordinate is measured in the direction of the flow. The '$_n$' coordinate direction always lies in the plane defined by the 's' coordinate direction and the fixed 'y' axis. The 'm' direction is normal to the (s,n) coordinate surface. The flow angles, θ and ϕ, are defined as shown in Fig. A1.

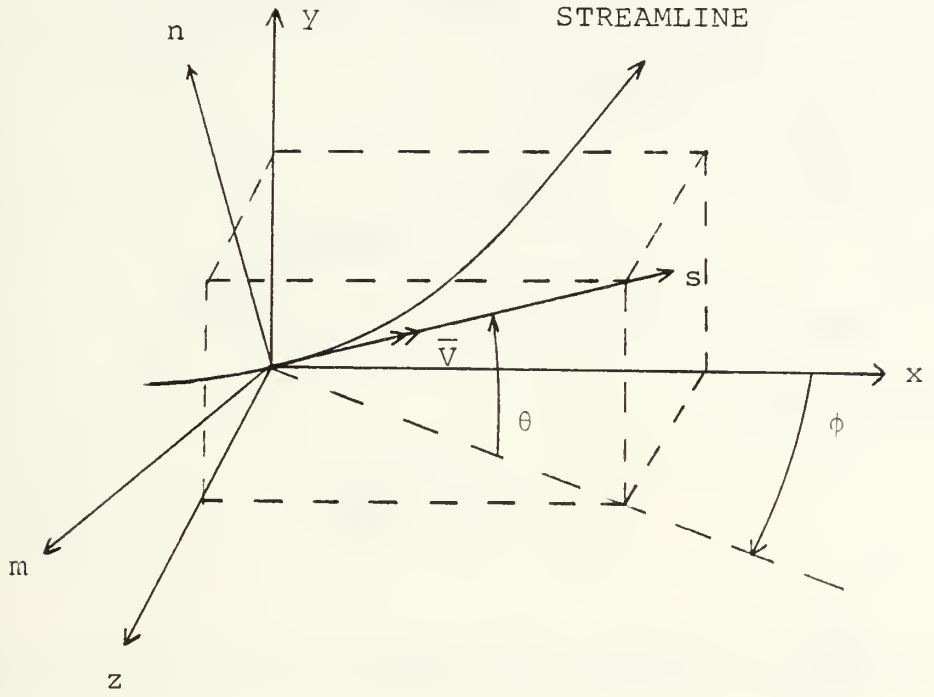


Figure A1. The Natural Coordinate System

C. VECTOR OPERATORS IN THE NATURAL COORDINATE SYSTEM

1. $\nabla(\ )$

If $d\vec{r}$ is the position vector of a fluid particle

$$d\vec{r} \cdot \nabla(\ ) = d(\ ) \tag{A1}$$

In natural coordinates, Eq. (A1) becomes

$$[\hat{i}_s ds + \hat{i}_n dn + \hat{i}_m dm] \cdot [(\ )\hat{i}_s + (\ )\hat{i}_n + (\ )\hat{i}_m]$$

$$= \frac{\partial(\ )}{\partial s} ds + \frac{\partial(\ )}{\partial n} dn + \frac{\partial(\ )}{\partial m} dm$$

By inspection,

$$\nabla(\ ) = \frac{\partial(\ )}{\partial s}\hat{i}_s + \frac{\partial(\ )}{\partial n}\hat{i}_n + \frac{\partial(\ )}{\partial m}\hat{i}_m \tag{A2}$$

2. $\underline{\bar{V}\cdot\nabla(\ )}$

Since

$$\bar{V} = q\,\hat{i}_s$$

using Eq. (A2)

$$\bar{V}\cdot\nabla(\ ) = q\,\frac{\partial(\ )}{\partial s} \tag{A3}$$

3. $\underline{\nabla\cdot\bar{V}}$

From Eq. (A2) with $\bar{V} = q\vec{i}_s$

$$\nabla\cdot\bar{V} = \frac{\partial q}{\partial s} \tag{A4}$$

4. $\underline{(\bar{V}\cdot\nabla)\bar{V}}$

From Eq. (A3) with $\bar{V} = q\vec{i}_s$

$$(\bar{V}\cdot\nabla)\bar{V} = q\,\frac{\partial\bar{V}}{\partial s} = q\,\frac{\partial(q\hat{i}_s)}{\partial s} = q\left[\frac{\partial q}{\partial s}\hat{i}_s + q\,\frac{\partial\hat{i}_s}{\partial s}\right] \tag{A5}$$

The change in the unit vector $\hat{i}_s$ with respect to s is derived below and is illustrated in Figs. A2 through A6. (A three-dimensional model is very helpful in visualizing the vector geometry.)

Figure A2 illustrates the natural coordinate system translating along a streamline from point A to point B in

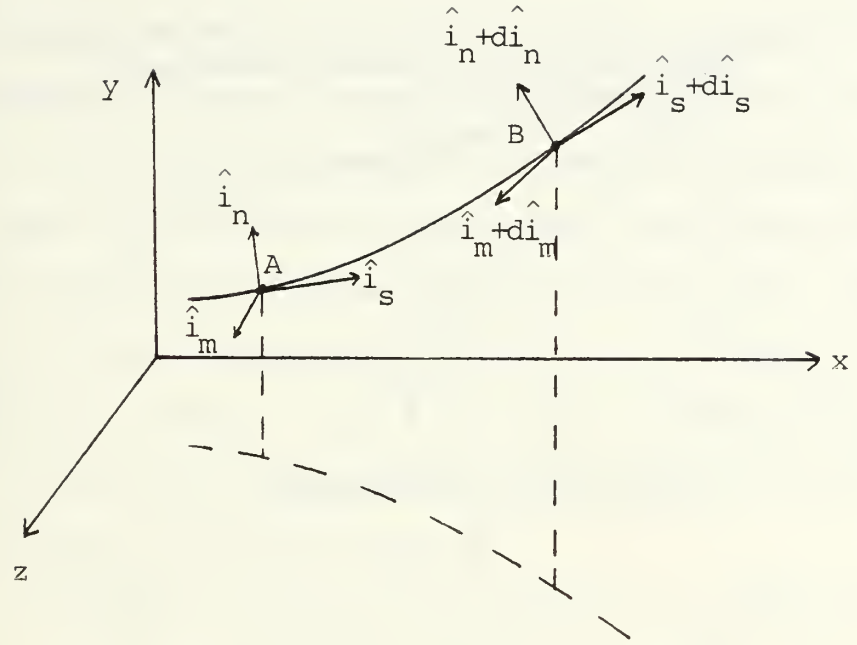


Figure A2. Unit Vectors Moving Along A Streamline

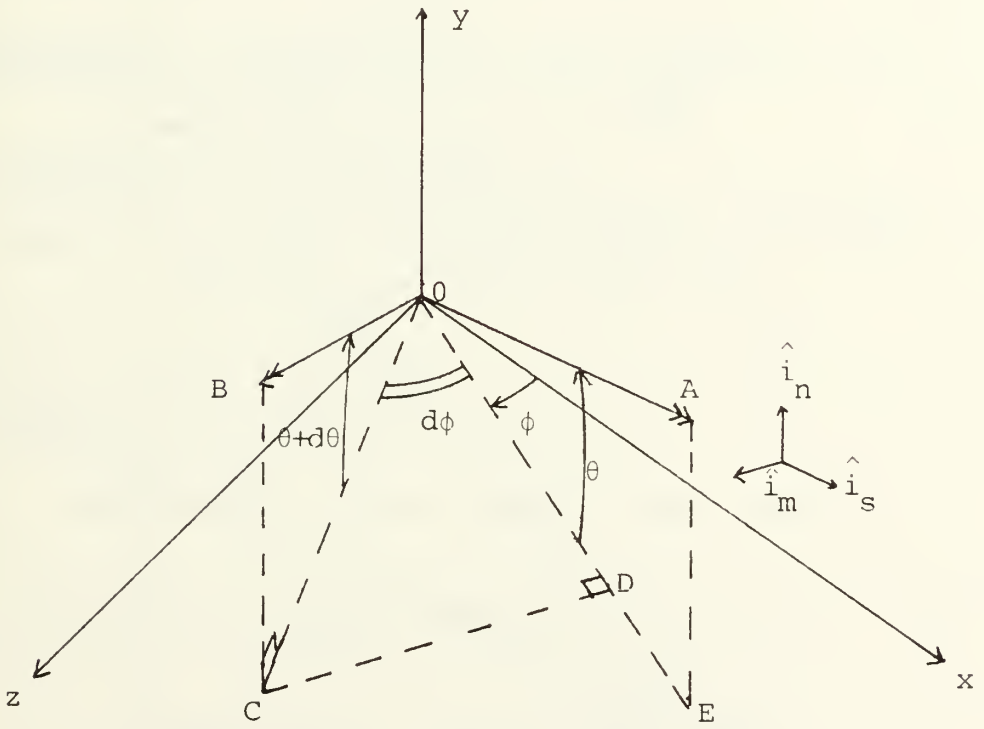


Figure A3. Superposition of $\hat{i}_s$ Vectors from Points A & B

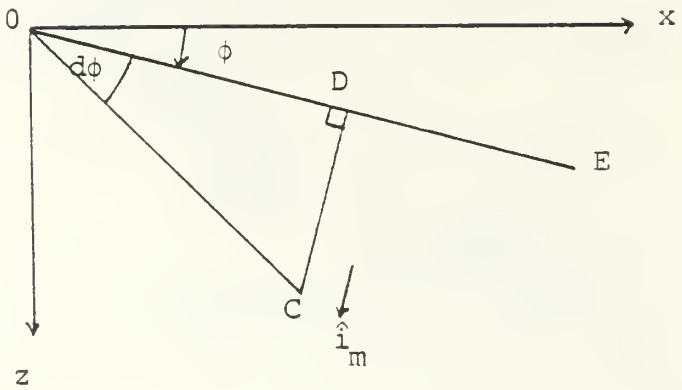


Figure A5. X-Z Plane

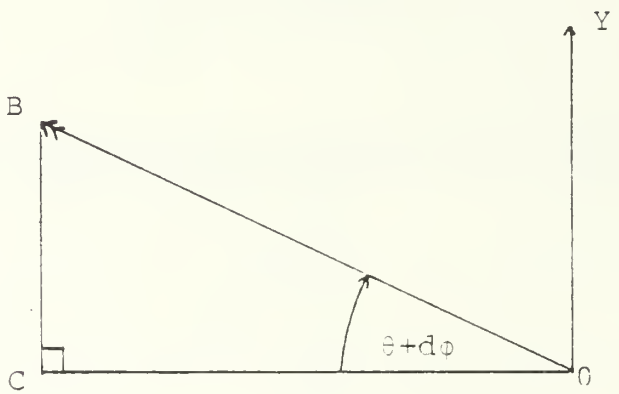


Figure A4. Plane Formed by B Vector and Y Axis

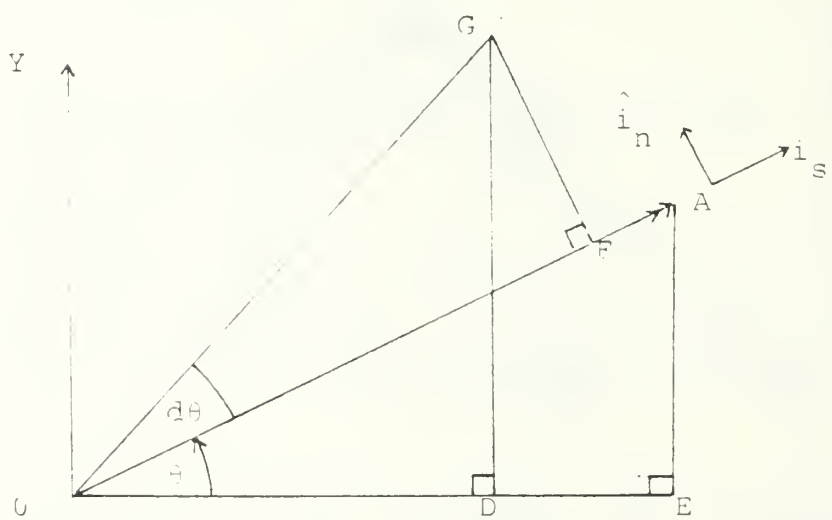


Figure A6. Plane Formed by A Vector and Y Axis

the inertial cartesian coordinate system. Figure A3 is the superposition of the $\hat{i}_s$ unit vector from points A and B of Fig. A2. The intent here is to express the change in the unit vector $\hat{i}_s$ in terms of components along the original (s,n,m) directions shown at point A in Fig. A3. Vector $\overline{OA}$ in Fig. A3 is the original $\hat{i}_s$ direction and vector $\overline{OB}$ is $\hat{i}_s + \frac{\partial \hat{i}_s}{\partial s} ds$. Points C and E are the projection of points B and A in the (x,z) plane respectively. Point D is the projection of point C onto the line $\overline{OE}$.

Figure A4 is the plane formed by the $\overline{OB}$ vector and the Y axis. The length of $\overline{OB}$ is unity by definition. The length of $\overline{BC}$ is $\sin(\theta+d\theta)$.

$$\sin(\theta+d\theta) = \sin\theta \cos d\theta + \cos\theta \sin d\theta$$

which, since $d\theta$ is a small angle, is

$$\sin(\theta+d\theta) = \sin\theta + \cos\theta \, d\theta \qquad \text{(A6)}$$

The length of $\overline{OC}$ is $\cos(\theta+d\theta)$.

$$\cos(\theta+d\theta) = \cos\theta \cos d\theta - \sin\theta \sin d\theta$$

which, since $d\theta$ is a small angle, is

$$\cos(\theta+d\theta) = \cos\theta - \sin\theta \, d\theta \qquad \text{(A7)}$$

Figure A5 is the (x,z) plane in which the angle ϕ is defined. The $\hat{i}_m$ direction is shown at point C. The length of $\overline{CD}$ is $\overline{OC}\sin d\phi$ which, with Eq. (A7) is

$$[\cos\theta - \sin\theta\, d\theta]\sin d\phi$$

which, since $d\phi$ is a small angle, is

$$\cos\theta\, d\phi - \sin\theta\, d\theta\, d\phi$$

Dropping the second order term

$$\overline{CD} = \cos\theta\, d\phi \qquad \text{(A8)}$$

The length of $\overline{OD}$ is $\overline{OC}\cos d\phi$ which, since $d\phi$ is a small angle, is $\overline{OC}$.

Figure A6 is the plane formed by the $\overline{OA}$ vector and the y axis. $\overline{GD}$ is the projection of $\overline{BC}$ and since both are vertical lines, $\overline{BC}$ and $\overline{GD}$ have the same length, which has been determined in Eq. (A6). The $\hat{i}_s$ and $\hat{i}_n$ directions are shown at point A. The lengths to be determined are $\overline{AF}$ and $\overline{GF}$, since these are the components of $\overline{AB}$ in the s and n directions. For small angles, $\overline{OD}$ was shown to be equal to $\overline{OC}$ and $\overline{GD} = \overline{BC}$, so that the angle GOD is equal to angle BOC = $\theta+d\theta$. Angle GOF is therefore $d\theta$, the length of $\overline{OG}$ is unity and the length of $\overline{GF}$ is sin $d\theta$, which for small angles is $d\theta$, and it is in the $\hat{i}_n$ direction. The length of $\overline{AF}$ is $\overline{OA} - \overline{OF}$. $\overline{OA}$ is unity

by definition and $\overline{OF}$ is cos $d\theta$, therefore $\overline{AF}$, the component of $\overline{AB}$ in the $\hat{i}_s$ direction is equal to $1 - \cos d\theta$, which, since $d\theta$ is a small angle, is zero.

The component of $\overline{AB}$ in the $\hat{i}_n$ direction is $\overline{GF}$ or

$$d\theta \, \hat{i}_n \tag{A9}$$

The remaining component of $\overline{AB}$ in the $\hat{i}_m$ direction is $\overline{CD}$ or

$$\cos \theta \, d\phi \, \hat{i}_m \tag{A10}$$

Therefore, the vector $\overline{AB}$ can be written as

$$\frac{\partial \hat{i}_s}{\partial s} ds = d\theta \, \hat{i}_m + \cos \theta \, d\phi \, \hat{i}_m$$

$$= \frac{\partial \theta}{\partial s} ds \, \hat{i}_n + \cos \theta \, \frac{\partial \phi}{\partial s} ds \, \hat{i}_m$$

and, dividing by ds, finally

$$\frac{\partial \hat{i}_s}{\partial s} = \frac{\partial \theta}{\partial s} \hat{i}_n + \cos \theta \, \frac{\partial \phi}{\partial s} \hat{i}_m \tag{A11}$$

Note that Fig. A3 can be considered also to describe a change in the unit vector $\hat{i}_s$ at a fixed location, with respect to time. In this case, the vector $\overline{AB}$ is equal to $\frac{\partial \hat{i}_s}{\partial t} dt$ and with Eq. (A9) and Eq. (A10)

$$\frac{\partial \hat{i}_s}{\partial t} = \frac{\partial \theta}{\partial t} \hat{i}_n + \cos\theta \frac{\partial \phi}{\partial t} \hat{i}_m \qquad \text{(A12)}$$

With Eq. (A11), Eq. (A5) becomes

$$(\bar{V}\cdot\nabla)\bar{V} = q \frac{\partial q}{\partial s} \hat{i}_s + q^2 \frac{\partial \theta}{\partial s} \hat{i}_n + q^2 \cos\theta \frac{\partial \phi}{\partial s} \hat{i}_m \qquad \text{(A13)}$$

D. CONSERVATION OF MASS

In the quasi-one-dimensional differential stream tube, shown in Figure A7, where $\hat{i}_s$ is always in the direction of $\bar{V}$, ρ, q, and the cross sectional area A are known at the center of the element and ρ and q are assumed constant on any given cross section.

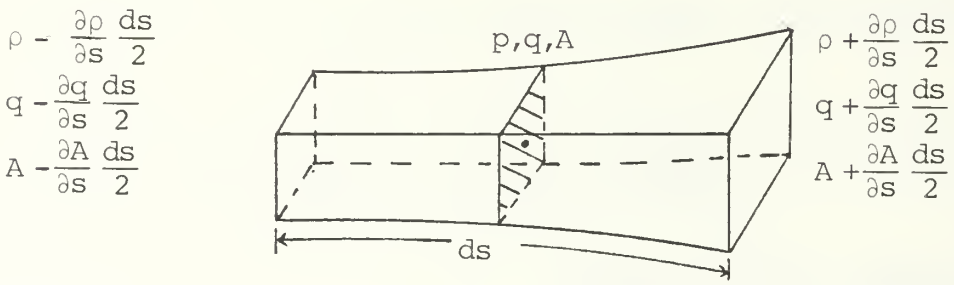


Figure A7. Differential Stream Tube of Variable Cross Section

The statement of continuity is

[The change in the mass within the control volume with respect to time] = [The net influx of mass across the control surface]

Since the volume is constant, the left side becomes

$$A\ ds\ \frac{\partial \rho}{\partial t}$$

For the stream tube, mass enters and leaves only from the ends so that the right hand side can be written as

$$[\rho - \frac{\partial \rho}{\partial s}\frac{ds}{2}][q - \frac{\partial q}{\partial s}\frac{ds}{2}][A - \frac{\partial A}{\partial s}\frac{ds}{2}] - [\rho + \frac{\partial \rho}{\partial s}\frac{ds}{2}][q + \frac{\partial q}{\partial s}\frac{ds}{2}][A + \frac{\partial A}{\partial s}\frac{ds}{2}]$$

which on expansion becomes

$$-\rho q \frac{\partial A}{\partial s} ds - qA \frac{\partial \rho}{\partial s} ds - \rho A \frac{\partial q}{\partial s} ds - \frac{\partial \rho}{\partial s}\frac{\partial q}{\partial s}\frac{\partial A}{\partial s} ds^3$$

On dropping the higher order terms and combining with the left hand side,

$$\frac{\partial \rho}{\partial t} + \rho q \frac{1}{A}\frac{\partial A}{\partial s} + q \frac{\partial \rho}{\partial s} + \rho \frac{\partial q}{\partial s} = 0 \qquad \text{(A14)}$$

Equation (A14) is the form of the continuity equation which is required to model a flow as being one-dimensional with area change. In general, however, $\frac{1}{A}\frac{\partial A}{\partial s}$ must be expressed in terms of q, θ and ϕ.

With reference to Fig. A8, the change in the cross-sectional area of the stream tube can be written as

$$\frac{\partial A}{\partial s} ds = [dm + \frac{\partial dm}{\partial s} ds][dn + \frac{\partial dn}{\partial s} ds] - A \qquad \text{(A15)}$$

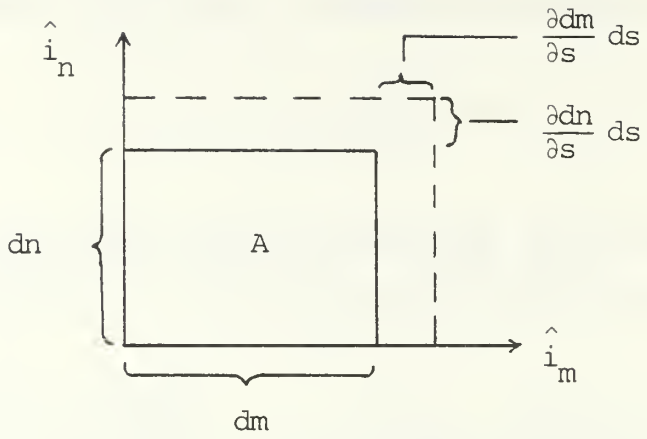


Figure A8. Cross Section of a Diverging Differential Stream Tube

After expansion, dividing by A and dropping the higher order terms, Eq. (A15) becomes

$$\frac{1}{A}\frac{\partial A}{\partial s} = \frac{1}{dn}\frac{\partial dn}{\partial s} + \frac{1}{dm}\frac{\partial dm}{\partial s} \tag{A16}$$

From Fig. A9,

$$\frac{1}{dn}\frac{\partial dn}{\partial s}\,ds = \frac{1}{dn}\,\frac{ds}{\cos(\frac{\partial \theta}{\partial n}\,dn)}\sin(\frac{\partial \theta}{\partial n}\,dn)$$

which, for small angles, becomes

$$\frac{1}{dn}\frac{\partial dn}{\partial s} = \frac{\partial \theta}{\partial n} \tag{A17}$$

From Fig. A10,

$$\frac{1}{dm}\frac{\partial dm}{\partial s}\,ds = \frac{1}{dm}\,\overline{AB} \tag{A18}$$

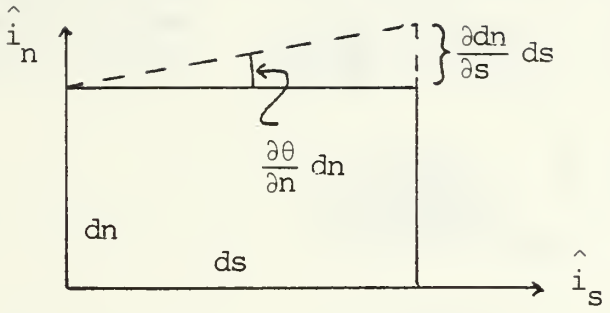


Figure A9. (s,n) Plane of Diverging Differential Stream Tube

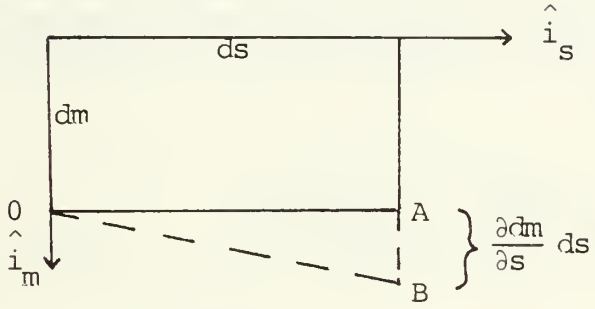


Figure A10. (s,m) Plane of Diverging Differential Stream Tube

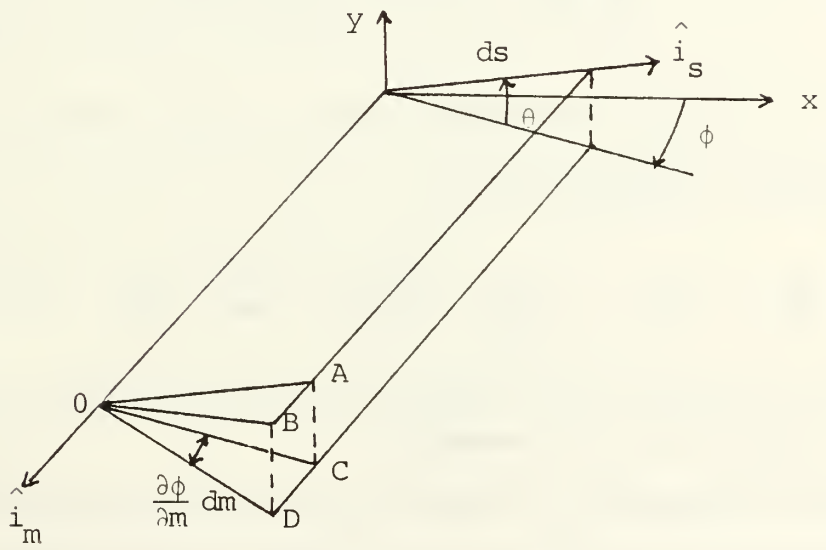


Figure A11. (s,m) Plane and its Projection onto the (x,z) Plane

With reference to Fig. A11, $\overline{AB} = \overline{CD}$, and

$$\overline{CD} = \overline{OD} \sin \left[\frac{\partial \phi}{\partial m} dm\right] \tag{A19}$$

$$\overline{OD} = \frac{\overline{OC}}{\cos \left[\frac{\partial \phi}{\partial m} dm\right]} \tag{A20}$$

$$\overline{OC} = ds \cos \theta \tag{A21}$$

Combining Eqs. (A18) through (A21) and knowing that $\frac{\partial \phi}{\partial m} dm$ is a small angle,

$$\frac{1}{dm} \frac{\partial dm}{\partial s} = \cos \theta \frac{\partial \phi}{\partial m} \tag{A22}$$

Equation (A16), with Eqs. (A17) and (A22), becomes

$$\frac{1}{A} \frac{\partial A}{\partial s} = \frac{\partial \theta}{\partial n} + \cos \theta \frac{\partial \phi}{\partial m} \tag{A23}$$

With Eq. (A23), the general form of the continuity equation in natural coordinates is

$$\frac{\partial \rho}{\partial t} + q \frac{\partial \rho}{\partial s} + \rho \frac{\partial q}{\partial s} + \rho q \left[\frac{\partial \theta}{\partial n} + \cos \theta \frac{\partial \phi}{\partial m}\right] = 0 \tag{A24}$$

E. CONSERVATION OF MOMENTUM

The statement of conservation for an arbitrary control volume which is fixed in the reference frame can be written as

$$\left[\begin{array}{l}\text{The change in the}\\ \text{momentum of the fluid}\\ \text{in the control volume}\\ \text{with respect to time}\end{array}\right] = \left[\begin{array}{l}\text{The net influx}\\ \text{of momentum}\\ \text{across the con-}\\ \text{trol surface}\end{array}\right] + \left[\begin{array}{l}\text{The net force}\\ \text{on the con-}\\ \text{trol surface}\end{array}\right]$$

$$+ \left[\begin{array}{l}\text{The net}\\ \text{body}\\ \text{force}\\ \text{on the fluid}\end{array}\right]$$

Assuming the fluid is inviscid, the only forces acting on the control surface are pressure forces. In the absence of any body forces (gravity, electromagnetic, etc.), the statement becomes

$$\frac{\partial}{\partial t}\iiint \rho\bar{V}\, dvol = -\iint \bar{V}\rho\bar{V}\cdot d\vec{a} - \iint P\, d\vec{a} = 0 \qquad (A25)$$

With Gauss' Theorem, Eq. (A25) becomes

$$\iiint \frac{\partial}{\partial t}[\rho\bar{V}]dvol + \iiint \nabla\cdot[\bar{V}\rho\bar{V}]dvol + \iiint \nabla P\, dvol = 0$$

or

$$\iiint\{\frac{\partial}{\partial t}[\rho\bar{V}] + \nabla\cdot[\bar{V}\rho\bar{V}] + \nabla P\}dvol = 0 \qquad (A26)$$

Since Eq. (A26) is true for any volume, the integrand must be zero, thus

$$\frac{\partial}{\partial t}[\rho\bar{V}] + \nabla\cdot[\bar{V}\rho\bar{V}] + \nabla P = 0 \qquad (A27)$$

Expanding the first term of Eq. (A27) and using the vector identity

$$\nabla\cdot[\bar{V}\rho\bar{V}] = \bar{V}[\nabla\cdot\rho\bar{V}] + \rho(\bar{V}\cdot\nabla)\bar{V}$$

Eq. (17) becomes

$$\rho\frac{\partial\bar{V}}{\partial t} + \bar{V}\{\frac{\partial\rho}{\partial t} + \nabla\cdot\rho\bar{V}\} + \rho(\bar{V}\cdot\nabla)\bar{V} + \nabla P = 0 \qquad \text{(A28)}$$

The second term of Eq. (A28) is zero by the conservation of mass so that Eq. (A28) reduces to

$$\rho\frac{\partial\bar{V}}{\partial t} + \rho(\bar{V}\cdot\nabla)\bar{V} + \nabla P = 0 \qquad \text{(A29)}$$

Using Eqs. (A2), (A5), (A11) and (A12), collecting terms and equating each vector component to zero, Eq. (A29) becomes

$$\hat{i}_s[\rho\frac{\partial q}{\partial t} + \rho q\frac{\partial q}{\partial s} + \frac{\partial P}{\partial s}] = 0 \qquad \text{(A30)}$$

$$\hat{i}_n[\rho q\frac{\partial\theta}{\partial t} + \rho q^2\frac{\partial\theta}{\partial s} + \frac{\partial P}{\partial n}] = 0 \qquad \text{(A31)}$$

$$\hat{i}_m[\rho q\cos\theta\frac{\partial\phi}{\partial t} + \rho q^2\cos\theta\frac{\partial\phi}{\partial s} + \frac{\partial P}{\partial m}] = 0 \qquad \text{(A32)}$$

Using the perfect gas relation

$$A^2 = \frac{\gamma P}{\rho}$$

and the identity

$$\frac{\partial P}{\partial s} = P \frac{\partial \ell n P}{\partial s}$$

Eqs. (A30) through (A32) become

$$\frac{\partial q}{\partial t} + q \frac{\partial q}{\partial s} = - \frac{A^2}{\gamma} \frac{\partial \ell n P}{\partial s} \tag{A33}$$

$$\frac{\partial \theta}{\partial t} + q \frac{\partial \theta}{\partial s} = - \frac{A^2}{\gamma q} \frac{\partial \ell n P}{\partial n} \tag{A34}$$

$$\frac{\partial \phi}{\partial t} + q \frac{\partial \phi}{\partial s} = - \frac{A^2}{\gamma q \cos \theta} \frac{\partial \ell n P}{\partial m} \tag{A35}$$

which are in the desired form.

F. CONSERVATION OF ENERGY

The conservation of energy for an arbitrary control volume is

$$\left[\begin{array}{l}\text{The rate of increase of energy within the control volume with respect to time}\end{array}\right] = \left[\begin{array}{l}\text{The rate at which energy is entering the control volume across the boundary}\end{array}\right] + \left[\begin{array}{l}\text{The net rate of work done on the fluid at the boundary}\end{array}\right] \tag{A36}$$

Neglecting gravitational potential, the energy per unit mass is

$$e + \frac{q^2}{2}$$

where e is the internal energy per unit mass and q is the velocity magnitude. For an inviscid fluid in the absence of body forces, the only work done on the fluid is pressure work. Thus Eq. (A36) is written mathematically as

$$\frac{\partial}{\partial t} \iiint [e + \frac{q^2}{2}] \rho \, dvol = -\iint [e + \frac{q^2}{2}] \rho \bar{V} \cdot d\vec{A} - \iint P\bar{V} \cdot d\vec{a} \qquad \text{(A37)}$$

Using Gauss' Theorem, Eq. (A37) is written as

$$\iiint \frac{\partial}{\partial t} [e + \frac{q^2}{2}] \rho \, dvol = - \iiint \nabla \cdot [(e + \frac{q^2}{2}) \rho \bar{V}] dvol$$

$$- \iiint \nabla \cdot [P\bar{V}] dvol$$

or

$$\frac{\partial}{\partial t} \{ [e + \frac{q^2}{2}] \rho \} + \nabla \cdot \{ [e + \frac{q^2}{2}] \rho \bar{V} \} = -\nabla \cdot [P\bar{V}]$$

Expanding,

$$\frac{\partial \rho}{\partial t} [e + \frac{q^2}{2}] + \rho \frac{\partial}{\partial t} [e + \frac{q^2}{2}] + (\rho \bar{V}) \nabla \cdot [e + \frac{q^2}{2}] + [e + \frac{q^2}{2}] [\nabla \cdot (\rho \bar{V})]$$

$$= -\nabla \cdot [P\bar{V}]$$

or

$$[e + \frac{q^2}{2}][\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{V})] + \rho \frac{\partial}{\partial t}[e + \frac{q^2}{2}] + (\rho \bar{V}) \nabla \cdot [e + \frac{q^2}{2}]$$

$$= -\nabla \cdot [P\bar{V}] \tag{A38}$$

The first term of Eq. (A38) is zero according to the conservation of mass. Expanding the remaining terms of Eq. (A38),

$$\rho \frac{\partial}{\partial t}[e + \frac{q^2}{2}] + \rho q \frac{\partial}{\partial x}[e + \frac{q^2}{2}] = -\nabla \cdot [P\bar{V}]$$

or

$$\frac{D}{Dt}[e + \frac{q^2}{2}] = -\frac{1}{\rho} \nabla \cdot [P\bar{V}] \tag{A39}$$

Equation (A39) can be written as

$$\frac{De}{Dt} + \frac{D}{Dt}[\frac{q^2}{2}] = -\frac{1}{\rho}(\bar{V} \cdot \nabla P) - \frac{P}{\rho}(\nabla \cdot \bar{V}) \tag{A40}$$

With $\bar{V} = q\hat{i}_s$ in the natural coordinate system, Eq. (A29) becomes

$$\frac{\partial \bar{V}}{\partial t} + q \frac{\partial \bar{V}}{\partial s} = -\frac{1}{\rho} \nabla P$$

or

$$\frac{D\bar{V}}{Dt} = -\frac{1}{\rho} \nabla P \tag{A41}$$

Taking the dot product of $\bar{V}$ with Eq. (A41),

$$\bar{V}\cdot\frac{D\bar{V}}{Dt} = -\frac{1}{\rho}[\bar{V}.\nabla P] \tag{A42}$$

and substituting Eq. (A42) into Eq. (A40)

$$\frac{De}{Dt} + \frac{D}{Dt}[\frac{q^2}{2}] = \bar{V}\cdot\frac{D\bar{V}}{Dt} - \frac{P}{\rho}(\nabla\cdot\bar{V}) \tag{A43}$$

But

$$\frac{D\bar{V}}{Dt} = \frac{D(q\hat{i}_s)}{Dt} = \frac{Dq}{Dt}\hat{i}_s + q\frac{D\hat{i}_s}{Dt}$$

$$= \frac{Dq}{Dt}\hat{i}_s + q[\frac{\partial\hat{i}_s}{\partial t} + q\frac{\partial\hat{i}_s}{\partial s}]$$

and using Eq. (A11) and Eq. (A12),

$$\frac{D\bar{V}}{Dt} = \frac{Dq}{Dt}\hat{i}_s + q[\frac{\partial\theta}{\partial t}\hat{i}_n + \cos\theta\frac{\partial\phi}{\partial t}\hat{i}_m]$$

$$+ q^2[\frac{\partial\theta}{\partial s}\hat{i}_n + \cos\theta\frac{\partial\phi}{\partial s}\hat{i}_m]$$

Therefore

$$\bar{V}\cdot\frac{D\bar{V}}{Dt} = q\hat{i}_s\cdot\{\frac{Dq}{Dt}\hat{i}_s + [q\frac{\partial\theta}{\partial t} + q^2\frac{\partial\theta}{\partial s}]\hat{i}_n + q\cos\theta[\frac{\partial\phi}{\partial t} + q\frac{\partial\phi}{\partial s}]\hat{i}_m\}$$

$$= q\frac{Dq}{Dt} = \frac{D}{Dt}[\frac{q^2}{2}]$$

Therefore Eq. (A43) reduces to

$$\frac{De}{Dt} = -\frac{P}{\rho}(\nabla\cdot\bar{V}) \tag{A44}$$

From the conservation of mass

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\bar{V}) = 0$$

or

$$\frac{\partial\rho}{\partial t} + \bar{V}\cdot\nabla\rho + \rho(\nabla\cdot\bar{V}) = 0$$

which, in the natural coordinate system gives,

$$\frac{\partial\rho}{\partial t} + q\,\frac{\partial\rho}{\partial s} = -\rho(\nabla\cdot\bar{V})$$

or

$$-\frac{1}{\rho}\,\frac{D\rho}{Dt} = (\nabla\cdot\bar{V}) \tag{A45}$$

Substituting Eq. (10) into Eq. (9),

$$\frac{De}{Dt} - \frac{P}{\rho^2}\,\frac{D\rho}{Dt} = 0 \tag{A46}$$

The definition of the modified entropy is

$$dS = -\frac{1}{\gamma}\,\frac{dQ_R}{T}$$

and the first law of thermodynamics for an elemental mass requires that

$$dQ_R = de + Pd\left(\frac{1}{\rho}\right)$$

so that

$$dS = -\frac{1}{\gamma T}\left[de + Pd\left(\frac{1}{\rho}\right)\right]$$

$$= -\frac{1}{\gamma T}\left[de + \frac{P}{\rho^2}\, d\rho\right]$$

and Eq. (A46) implies that

$$\frac{DS}{Dt} = 0 \qquad \text{(A47)}$$

which states that modified entropy is conserved along streamlines

G. MODIFIED ENTROPY EVALUATION

With the modified entropy defined as

$$dS = -\frac{dQ_R}{\gamma T}$$

or

$$S_B - S_A = -\frac{1}{\gamma}\int_A^B \frac{dQ_R}{T} \qquad \text{(A48)}$$

From the first law of thermodynamics

$$dQ_R = de + P\,dv \tag{A49}$$

where e is the internal energy and v is the specific volume. Substituting Eq. (A49) into Eq. (A48),

$$S_A - S_B = \frac{1}{\gamma}\int_A^B \frac{de + P\,dv}{T} = \frac{1}{\gamma}\left[\int_A^B \frac{de}{T} + \int_A^B \frac{P\,dv}{T}\right] \tag{A50}$$

For a perfect gas

$$P\,v = R_G T \qquad \text{and} \qquad de = C_v\,dT$$

for which Eq. (A50) becomes

$$S_A - S_B = \frac{1}{\gamma}\left[\int_A^B \frac{C_v\,dT}{T} + \int_A^B \frac{R_G\,dv}{v}\right]$$

which after integration yields

$$S_A = S_B + \frac{1}{\gamma}\left[C_v \ln \frac{T_B}{T_A} + R \ln \frac{v_B}{v_A}\right] \tag{A51}$$

Substituting

$$R_G = C_v(\gamma - 1)$$

into Eq. (A51)

$$S_A = S_B + \frac{1}{\gamma}\left[C_v\left(\ell n \frac{T_B}{T_A} - \ell n \frac{v_B}{v_A}\right) + C_v \gamma \ell n \frac{v_B}{v_A}\right]$$

$$S_A = S_B + \frac{1}{\gamma}\left[\frac{R_G}{\gamma-1} \ell n \left(\frac{P_B v_B R_A v_A}{P_A v_A R_B v_B}\right) + \frac{\gamma R}{\gamma-1} \ell n \frac{\rho_A}{\rho_B}\right]$$

and since $R_{G_A} = R_{G_B}$

$$S_A = S_B + \frac{R}{\gamma(\gamma-1)}\left[\ell n \frac{P_B}{P_A} - \gamma \ell n \frac{\rho_B}{\rho_A}\right]$$

or

$$\frac{S_A}{R_G} = \frac{S_B}{R_G} + \frac{1}{\gamma(\gamma-1)}\left[\ell n \frac{P_B}{\rho_B \gamma} - \ell n \frac{P_A}{\rho_A \gamma}\right]$$

or

$$\frac{S_B}{R_G} = \frac{S_A}{R_G} + \frac{1}{\gamma(\gamma-1)}\left[\ell n \frac{P_A}{\rho_A \gamma} - \ell n \frac{P_B}{\rho_B \gamma}\right] \qquad \text{(A52)}$$

If the reference condition is chosen to be

$$\frac{S_A}{R_G} = \frac{2}{\gamma-1} - \frac{1}{\gamma(\gamma-1)} \ell n \frac{P_A}{\rho_A \gamma}$$

Eq. (A52) becomes

$$\frac{S_B}{R_G} = \frac{2}{\gamma-1} - \frac{1}{\gamma(\gamma-1)} \ell n \frac{P_B}{\rho_B \gamma} \qquad \text{(A53)}$$

H. FURTHER TRANSFORMATIONS

Equations (A24), (A33) through (A35) and (A47) are in the required form. However, some of the variables need to be transformed, namely, derivatives of $\ell n\ P$ need to be expressed in terms of A and S and derivatives of ρ must be expressed in terms of q and A.

Using the first law of thermodynamics expressed as

$$dQ = dh - \frac{1}{\rho} dP$$

Eq. (A48) may be written as

$$-\gamma\ T \nabla S = \nabla h - \frac{1}{\rho} \nabla P$$

or for the $\hat{i}_s$ component

$$-\gamma \frac{\partial S}{\partial s} = \frac{1}{T} \frac{\partial C_p T}{\partial s} - \frac{P}{\rho T} \frac{\partial \ell n\ P}{\partial s} \qquad \text{(A54)}$$

Assuming C_p to be constant and using the equation of state for a perfect gas Eq. (A54) becomes

$$-\gamma \frac{\partial S}{\partial s} = \frac{C_p}{T} \frac{\partial}{\partial s}\left[\frac{P}{R_G \rho}\right] - R \frac{\partial\ \ell n\ P}{\partial s}$$

and since $C_p = \gamma R/(\gamma-1)$

$$-\gamma \frac{\partial S}{\partial s} = \frac{\gamma R_G}{(\gamma-1)} \frac{\rho R_G}{P} \frac{\partial}{\partial s}\left[\frac{P}{R_G \rho}\right] - R_G \frac{\partial\ \ell n\ P}{\partial s}$$

Since R is a constant it may be moved in or out of the derivatives giving

$$-\gamma \frac{\partial}{\partial s}\left[\frac{S}{R_G}\right] = \frac{\gamma}{\gamma-1} \frac{\rho}{P} \frac{\partial}{\partial s}\left[\frac{P}{\rho}\right] - \frac{\partial \ \ell n \ P}{\partial s}$$

and since γ is also constant, similarly

$$-\gamma \frac{\partial}{\partial s}\left[\frac{S}{R_G}\right] = \frac{\gamma}{\gamma-1} \frac{\rho}{P\gamma} \frac{\partial}{\partial s}\left[\frac{P\gamma}{\rho}\right] - \frac{\partial \ \ell n \ P}{\partial s}$$

Using $A^2 = \gamma P/\rho$

$$-\gamma \frac{\partial}{\partial s}\left[\frac{S}{R_G}\right] = \frac{\gamma}{\gamma-1} \frac{1}{A^2} \frac{\partial A^2}{\partial s} - \frac{\partial \ \ell n \ P}{\partial s}$$

$$= \frac{2A\gamma}{(\gamma-1)A^2} \frac{\partial A}{\partial s} - \frac{\partial \ \ell n \ P}{\partial s}$$

and, on rearranging

$$\frac{\partial \ \ell n \ P}{\partial s} = \frac{2\gamma}{\gamma-1} \frac{1}{A} \frac{\partial A}{\partial s} + \gamma \frac{\partial}{\partial s}\left[\frac{S}{R}\right] \qquad \text{(A55)}$$

Substituting Eq. (A55) into Eq. (A33) yields

$$\frac{\partial q}{\partial t} + q \frac{\partial q}{\partial s} + \frac{2A}{\gamma-1} \frac{\partial A}{\partial s} + A^2 \frac{\partial}{\partial s}\left[\frac{S}{R}\right] = 0 \qquad \text{(A56)}$$

and the derivative of ℓn P has been removed.

Since

$$A^2 = \frac{\gamma P}{\rho}$$

logarithmic differentiation yields

$$2\frac{dA}{A} = \frac{dP}{P} + \cancelto{0}{\frac{d\gamma}{\gamma}} - \frac{d\rho}{\rho} \tag{A57}$$

For a pure substance

$$P = P(\rho, S)$$

and an isentropic process

$$P = P(\rho)_S$$

Differentiating

$$\left(\frac{\partial P}{\partial \rho}\right)_S = \frac{dP}{d\rho} = \frac{\gamma P}{\rho} = A^2$$

therefore

$$\frac{dP}{P} = \gamma\frac{d\rho}{\rho} \tag{A58}$$

Substituting Eq. (A58) into Eq. (A57) gives

$$2\frac{dA}{A} = \gamma\frac{d\rho}{\rho} - \frac{d\rho}{\rho} = \gamma - 1\ \frac{d\rho}{\rho}$$

or

$$d\rho = \frac{2\rho}{(\gamma-1)} \frac{dA}{A} \tag{A59}$$

Along a streamline

$$d\rho = \frac{\partial \rho}{\partial t} + q \frac{\partial \rho}{\partial s} \tag{A60}$$

and since the fluid is the same from streamline to streamline Eqs. (A59) and Eq. (A60) can be combined giving

$$\frac{\partial \rho}{\partial t} + q \frac{\partial \rho}{\partial s} = \frac{2\rho}{\gamma-1} \frac{dA}{A} = \frac{2\rho}{(\gamma-1)A}[\frac{\partial A}{\partial t} + q \frac{\partial A}{\partial s}] \tag{A61}$$

Substituting Eq. (A61) into Eq. (A24) yields

$$\frac{\partial A}{\partial t} + q \frac{\partial A}{\partial s} + \frac{\gamma-1}{2} A \frac{\partial q}{\partial s} + q A \frac{\gamma-1}{2}[\frac{\partial \theta}{\partial n} + \cos\theta \frac{\partial \phi}{\partial m}] = 0 \tag{A62}$$

and the derivatives of ρ have been removed.

The governing equations are now summarized:

$$\frac{\partial A}{\partial t} + q \frac{\partial A}{\partial s} + \frac{\gamma-1}{2} A \frac{\partial q}{\partial s} + q A \frac{\gamma-1}{2}[\frac{\partial \theta}{\partial n} + \cos\theta \frac{\partial \phi}{\partial m}] = 0 \tag{A63}$$

$$\frac{\partial q}{\partial t} + q \frac{\partial q}{\partial s} + \frac{2A}{\gamma-1} \frac{\partial A}{\partial s} + A^2 \frac{\partial}{\partial s}[\frac{S}{R}] = 0 \tag{A64}$$

$$\frac{\partial \theta}{\partial t} + q \frac{\partial \theta}{\partial s} = - \frac{A^2}{\gamma q} \frac{\partial \ln P}{\partial n} \tag{A65}$$

$$\frac{\partial \phi}{\partial t} + q \frac{\partial \phi}{\partial s} = - \frac{A^2}{\gamma q \cos\theta} \frac{\partial \ln P}{\partial m} \tag{A66}$$

$$\frac{\partial S}{\partial t} + q \frac{\partial S}{\partial s} = 0 \qquad \text{(A67)}$$

Note that Eqs. (A65) through (A67) are in the form

$$\frac{\partial X}{\partial t} + \lambda \frac{\partial X}{\partial s} = Z$$

If $X = X(s,t)$, then

$$dX = \frac{\partial X}{\partial t} dt + \frac{\partial X}{\partial s} ds$$

and

$$\frac{dX}{dt} = \frac{\partial X}{\partial t} + \frac{ds}{dt} \frac{\partial X}{\partial s} \qquad \text{(A68)}$$

where (ds/dt) is the "characteristic" direction in the (s,t) plane along which Eq. (A68) holds. Along this direction, the equations may be transformed from partial to ordinary differential equations. What is desirable is to find transformed variables in terms of which Eqs. (A63) and (A64) will be in the same characteristic form. For this reason the modified Riemann variables

$$Q = q + AS/R_G \qquad \text{(A69)}$$

$$R = q - AS/R_G \qquad \text{(A70)}$$

are defined, where S/R_G is defined in Eq. (A53). The modified Riemann variables can be introduced by manipulation of the equations as follows. First, multiplying Eq. (A63) by S, gives

$$S \frac{\partial A}{\partial t} + q S \frac{\partial A}{\partial s} + \frac{\gamma-1}{2} A S \frac{\partial q}{\partial s} + q A S \frac{\gamma-1}{2}[\frac{\partial \theta}{\partial n} + \cos \theta \frac{\partial \phi}{\partial m}] = 0 \tag{A71}$$

and multiplying Eq. (A67) by A gives

$$A \frac{\partial S}{\partial t} + q A \frac{\partial S}{\partial s} = 0 \tag{A72}$$

On adding Eqs. (A71) and (A72) to Eq. (A64), and introducing

$$A \frac{\partial q}{\partial s} - A \frac{\partial q}{\partial s} = 0 \tag{A73}$$

and

$$A S \frac{\partial A}{\partial s} - A S \frac{\partial A}{\partial s} = 0 \tag{A74}$$

after appropriate rearrangement

$$\frac{\partial q}{\partial t} + A \frac{\partial S}{\partial t} + S \frac{\partial A}{\partial t} + q \frac{\partial q}{\partial s} + q A \frac{\partial S}{\partial s} + q S \frac{\partial A}{\partial s} + A \frac{\partial q}{\partial s}$$

$$+ A^2 \frac{\partial S}{\partial s} + A S \frac{\partial A}{\partial s} = - \frac{\gamma-1}{2} A S \frac{\partial q}{\partial s} + A S \frac{\partial A}{\partial s} + A \frac{\partial q}{\partial s}$$

$$- \frac{2A}{\gamma-1} \frac{\partial A}{\partial s} - \frac{\gamma-1}{2} q A S[\frac{\partial \theta}{\partial n} + \cos \theta \frac{\partial \phi}{\partial m}] = 0$$

Rearranging and introducing Eq. (A69) gives

$$\frac{\partial Q}{\partial t} + (q+A)\frac{\partial Q}{\partial s} = -\frac{\gamma-1}{2} A \left[S - \frac{2}{\gamma-1}\right]\left[\frac{\partial}{\partial s}\left(q - \frac{2}{\gamma-1}A\right)\right]$$

$$- \frac{\gamma-1}{2} q A S \left[\frac{\partial \theta}{\partial n} + \cos\theta \frac{\partial \phi}{\partial m}\right] \qquad \text{(A75)}$$

By subtracting Eqs. (A71) and (A72) from Eq. (A64), again using Eq. (A73) and Eq. (A74), and introducing Eq. (A70), one also obtains

$$\frac{\partial R}{\partial t} + (q-A)\frac{\partial R}{\partial s} = \frac{\gamma-1}{2} A \left[S - \frac{2}{\gamma-1}\right]\left[\frac{\partial}{\partial s}\left(q + \frac{2}{\gamma-1} A\right)\right]$$

$$+ \frac{\gamma-1}{2} q A S \left[\frac{\partial \theta}{\partial n} + \cos\theta \frac{\partial \phi}{\partial m}\right] \qquad \text{(A76)}$$

Together with Eqs. (A65) through (A67), Eqs. (A75) and (A76) form the system of equations which describe the isentropic flow of an inviscid perfect gas under unsteady, compressible conditions, and which may be solved as a system of ordinary differential equations along characteristic trajectories in the space-time domain.

I. SHOCK JUMP EQUATION

The analytical expression for the change in the extended Riemann variable, Q, across a stationary shock is derived below.

Consider a shock moving with a velocity V_s into a fluid with velocity q_B as depicted in Fig. A12, with the conditions

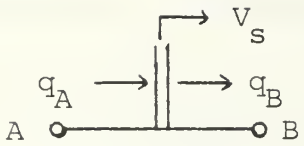


Figure A12

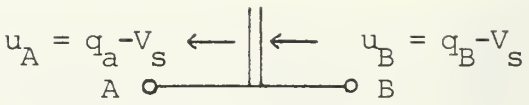


Figure A13

to the left of the shock being denoted by A subscripts and conditions to the right by B subscripts. If a velocity of $-V_s$ is imposed on the entire system, the situation becomes one of a stationary shock, as depicted in Fig. A13. In both cases, the high pressure side is to the left (A side). Since all velocities are defined positive to the right, a positive value for the relative mach number is defined as

$$W = -\frac{u_B}{A_B} = -\frac{(q_B - V_s)}{A_B} \tag{A77}$$

The extended Riemann variable, Q, is defined as

$$Q \equiv q + AS \tag{A78}$$

where

$$S \equiv \frac{2}{\gamma-1} - \frac{1}{\gamma(\gamma-1)} \ell n\left[\frac{p}{\rho^\gamma}\right] \tag{A79}$$

If all velocities are non-dimensionalized by A_B and pressures and densities are non-dimensionalized by their values at B,

$$\frac{Q_A - Q_B}{A_B} = \left[\frac{q_A + A_A S_A}{A_B}\right] - \left[\frac{q_B + A_B S_B}{A_B}\right]$$

$$\frac{Q_A - Q_B}{A_B} = \frac{q_A - q_B}{A_B} + \frac{A_A}{A_B} S_A - S_B$$

$$= \frac{q_A - q_B}{A_B} + \frac{A_A}{A_B}\left[\frac{2}{\gamma-1} - \frac{1}{\gamma(\gamma-1)} \ln \frac{p_A}{p_B}\left(\frac{\rho_A}{\rho_B}\right)^{\gamma}\right]$$

$$- \left[\frac{2}{\gamma-1} - \frac{1}{\gamma(\gamma-1)} \ln \frac{p_B}{p_B}\left(\frac{\rho_B}{\rho_B}\right)^{\gamma}\right]$$

$$= \frac{q_A - q_B}{A_B} + \left[\frac{A_A}{A_B} - 1\right]\frac{2}{\gamma-1} - \left(\frac{A_A}{A_B}\right)\frac{1}{\gamma(\gamma-1)}\left[\ln \frac{p_A}{p_B}\left(\frac{\rho_B}{\rho A}\right)^{\gamma}\right] \tag{A80}$$

The ratios of pressure, density and sound speed across a normal shock are well known functions of γ and the Mach number which in this case is W. They are

$$\frac{p_A}{p_B} = \frac{2\gamma}{\gamma+1} W^2 - \frac{\gamma-1}{\gamma+1} \tag{A81}$$

$$\frac{\rho_A}{\rho_B} = \frac{(\gamma+1) W^2}{(\gamma-1) W^2 + 2} \tag{A82}$$

$$\frac{A_A}{A_B} = \frac{1}{(\gamma+1) W}\left\{2(\gamma-1)\left[1 + \frac{\gamma-1}{2} W^2\right]\left[\frac{2\gamma}{\gamma-1} W^2 - 1\right]\right\}^{1/2} \tag{A83}$$

An expression for $(q_A - q_B)/A_B$ is obtained using mass conservation.

$$\rho_A u_A = \rho_B u_B$$

$$\frac{u_A}{u_B} = \frac{\rho_B}{\rho_A} = \frac{(\gamma-1)W^2 + 2}{(\gamma+1)W^2}$$

Subtracting one from both sides,

$$\frac{u_A - u_B}{u_B} = \frac{(\gamma-1)W^2 + 2 - (\gamma+1)W^2}{(\gamma+1)W^2}$$

$$\frac{u_A - u_B}{B} = \frac{2(1 - W^2)}{(\gamma+1)W^2} \qquad \text{(A84)}$$

Substituting Eq. (A77) for u_B in Eq. (A84) gives

$$\frac{u_A - u_B}{-WA_B} = \frac{2(1 - W^2)}{(\gamma+1)W^2}$$

$$\frac{u_A - u_B}{A_B} = \frac{2(W^2 - 1)}{(\gamma+1)W} \qquad \text{(A85)}$$

Since $u_A = q_A - V_s$ and $u_B = q_B - V_s$

$$u_A - u_B = q_A - V_s - q_B + V_s = q_A - q_B$$

Thus Eq. (A85) can be written as

$$\frac{q_A - q_B}{A_B} = \frac{2(W^2 - 1)}{(\gamma+1)W} \tag{A86}$$

Substituting Eqs. (A86) and (A81) through (A83) into Eq. (A80) gives

$$\frac{Q_A - Q_B}{A_B} = \frac{2(W^2 - 1)}{(\gamma+1)\ W} + \frac{2}{\gamma-1}\{\frac{1}{(\gamma+1)W}[2(\gamma-1)(1 + \frac{\gamma-1}{2}W^2)(\frac{2\gamma}{\gamma-1}W^2-1)]^{1/2}$$

$$- 1\} - \{\frac{1}{(\gamma+1)W}[2(\gamma-1)(1 + \frac{\gamma-1}{2}W^2)(\frac{2\gamma}{\gamma-1}W^2-1)]^{1/2}\}$$

$$\times \{\frac{1}{\gamma(\gamma-1)}[\ln(\frac{2\gamma}{\gamma+1}W^2 - \frac{\gamma-1}{\gamma+1})(\frac{(\gamma-1)W^2+2}{(\gamma+1)W^2})^{\gamma}\} \tag{A87}$$

APPENDIX B

USING EULER-1 ON THE NPS VM/CMS SYSTEM

A. MEMORY REQUIREMENTS

Storage should be defined as one mega-byte to ensure adequate memory for DISSPLA requirements. Also ensure that room exists on the user's disk for the listing file if that output option is selected.

B. TERMINAL REQUIREMENTS

The type of terminal required to run EULER-1 depends on the output option selected. When using the tabular listing output, any terminal tied into the VM/CMS system may be used. Graphical output may also be created using any terminal and the graph stored on the user's disk in a metafile for later viewing at a graphics terminal or for printing. To have the graph stored on the disk, use the "COMPRS" command on line 223 of the program and comment out the "TEK618" command on line 224. When graphical output is selected and it is desired to view the graph at the terminal at execution time, an IBM 3277-Tek618 dual screen terminal must be used. To use this option, use the "TEK618" command on line 224 and comment out the "COMPRS" command on line 223. When using this option, a low quality hard copy of the TEK618 display may be obtained using the TEK4631 hard copy unit attached.

C. PROCEDURE AND COMMANDS

After loging on to VM/CMS, use the command "DEFINE STORAGE 1M" to increase the virtual memory as recommended above. This will remove you from the CMS environment. Re-enter CMS with the command "I CMS."

Edit the user input area of EULER-1 FORTRAN as desired for the particular problem being run. When graphical output is selected, comment out either the "COMPRS" or the "TEK618" command as desired.

Compute the program using the command "FORTVS EULER-1." After compilation, an EULER-1 TEXT and an EULER-1 LISTING file will reside on the user's disk. At execution, if a tabular output has been selected, it will be sent to the user's disk as "FILE FT09F001." To give this file a particular name, say "EULER-1 LISTING," use the file definition command

"F I 9 DISK EULER-1 LISTING A (PERM"

at compile time. An alternate and convenient means of compiling the program is to set up an exec file on the user's disk by the name "EULER-1 EXEC" which contains the commands

```
F I 9 DISK EULER-1 LISTING A ( PERM
FORTVS EULER-1
```

Then to compile the program and define the output file, use the command "EULER-1."

After compilation, to execute the program, use the command "DISSPLA EULER-1." The message

...YOUR FORTRAN PROGRAM IS NOW BEING LOADED...

...EXECUTION WILL SOON FOLLOW...

should appear, followed shortly by

...EXECUTION BEGINS...

If tabular output was selected, proper termination will result in a ready message. If graphical output was selected using the TEK618 option, the plot will appear on the TEK618 terminal, the program will pause, and a

"...press ENTER to continue"

message will be displayed on the 3277 terminal. At this point the hard copy must be made, if desired, using the TEK4631 hard copy unit. After pressing the ENTER key on the 3277 terminal, the plot will be erased and the program will terminate. Proper termination will be indicated by the message

"END OF DISSPLAY 9.2 #### VECTORS IN 1 PLOT..."

and a ready message.

```
C     ************************************************                EUL00010
C     *                                              *                EJL00020
C     *                E U L E R - 1                 *                EUL00030
C     *                                              *                EUL00040
C     *                                              *                EUL00050
C     *    THIS PROGRAM SOLVES THE EULER EQUATIONS   *                EJL00060
C     *   EXPRESSED IN A QUAZI-ONE DIMENSIONAL       *                EUL00070
C     *   STREAMLINE COORDINATE SYSTEM.              *                EUL00080
C     *                                              *                EUL00090
C     *   AUTHOR - LT.  T.F. SALACKA                 *                EUL00100
C     *   LATEST REVISION - 02 DECEMBER  1985        *                EJL00110
C     *                                              *                EUL00120
C     *   FEATURES OF THIS VERSION (-1)              *                EUL00130
C     *   + ORDER OF SPATIAL DERIVITIVES - FIRST     *                EUL00140
C     *   + NUMBER OF SPATIAL DIMENSIONS - ONE       *                EUL00150
C     *   + DISCONTINUITIES TREATED:                 *                EUL00160
C     *         SHOCKS                 - YES         *                EUL00170
C     *         CONTACT DISCONTINUITIES - NO          *                EUL00180
C     *         EXPANSION WAVES        - NO          *                EJL00190
C     *                                              *                EUL00200
C     ************************************************                EJL00210
C                                                                      EJL00220
C     ++++++++++++++++++++++++++++++++++++++++++++++++                 EUL00230
C     +                                              +                 EJL00240
C     +      CONVENTIONS   AND   DEFINITIONS         +                 EJL00250
C     +                                              +                 EJL00260
C     ++++++++++++++++++++++++++++++++++++++++++++++++                 EUL00270
C                                                                      EJL00280
C     ------  NON-DIMENSIONING CONVENTION----------                    EJL00290
C                                                                      EJL00300
C       ALL VELOCITIES NON-DIM. BY THE SOUND SPEED ON                  EUL00310
C           THE LOW PRESSURE SIDE OF THE DIAPHRAGM.                    EJL00320
C       ALL PRESSURES, DENSITIES, AND TEMPERATURES ARE                 EUL00330
C           NON-DIM. BY THEIR INITIAL VALUES ON THE LOW                EUL00340
C           PRESSURE SIDE OF THE DIAPHRAGM.                            EUL00350
C       SPACIAL DISTANCE IS NON-DIM.BY OVERALL LENGTH.                 EUL00360
C       ENTROPY IS NON-DIM. BY THE GAS CONSTANT, R.                    EUL00370
C       TIME IS NON-DIM. BY (LENGTH/SOUND SPEED).                      EJL00380
C       VELOCITIES AND DISTANCES ARE DEFINED POSITIVE                  EUL00390
C           TO THE RIGHT.                                              EJL00400
C                                                                      EJL00410
C     ---------   SUBSCRIPT NOTATION-----------------                 EUL00420
C                                                                      EUL00430
C       I - SPACIAL NODE  (1 TO N FROM LEFT TO RIGHT)                  EJL00440
C       J - TIME LEVEL    (0 IS THE INITIAL CONDITION)                 EUL00450
C       K - DENOTES WHICH CHARACTERISTIC WAVE IS BEING                 EUL00460
C           DEALT WITH:                                                EJL00470
C                1 = Q+A     2 = Q      3 = Q-A                        EUL00480
```

```
C       L - DENOTES WHICH TYPE OF DISCONTINUITY IS                      EUL00490
C           BEING DEALT WITH:                                           EUL00500
C              1 = SHOCK    2 = CONTACT DISCONTINUITY                   EJL00510
C              3 = HEAD OF EXPANSION WAVE                               EUL00520
C              4 = TAIL OF EXPANSION WAVE                               EUL00530
C                                                                       EJL00540
C     -----------  VARIABLE DEFINITIONS  ------------                   EUL00550
C                                                                       EJL00560
C     A       - SPEED OF SOUND                                          EUL00570
C     COUNT   - COUNTER FOR GRAPHICS ROUTINES                           EUL00580
C     DARRAY  - ARRAY OF DENSITY FOR PLOTTING                           EUL00590
C     DELT    - TIME STEP                                               EUL00600
C     DLCD    - DENSITY TO THE LEFT OF THE CONTACT                      EUL00610
C                DISCONTINUITY IN THE EXACT SOLUTION.                   EUL00620
C     DLSH    - DENSITY TO THE LEFT OF THE SHOCK IN THE                 EUL00630
C                EXACT SOLUTION.                                        EUL00640
C     DRI     - INITIAL DENSITY RATIO ACROSS THE SHOCK                  EUL00650
C     EE      - DESIRED PRECISION FOR CHARACTERISTIC CALCULATIONS       EUL00660
C     G       - GAMMA (RATIO OF SPECIFIC HEATS)                         EUL00670
C     GRAPHS  - FOR GRAPHICAL OUTPUT,  0=NONE (TABULAR)                 EUL00680
C                1=PLOTS ALL VARIABLES                                  EUL00690
C                2=COMPARES DENSITY WITH EXACT SOLUTION                 EUL00700
C     G1      - 1/(G-1)                                                 EUL00710
C     G2      - 2/(G-1)                                                 EUL00720
C     H       - 1/(N-1)                                                 EUL00730
C     I2      - NODE # UPSTREAM OF A DISCONTINUITY                      EUL00740
C     JSTOP   - NUMBER OF TIME LEVELS TO BE CALCULATED                  EUL00750
C     M       - A NUMBER WHICH INDICATES WHICH                          EUL00760
C                DISCONTINUITIES WILL BE TREATED:                       EUL00770
C                1 - SHOCK ONLY                                         EJL00780
C                2 - SHOCK AND CONTACT DISCONTINUITY                    EUL00790
C     N       - NUMBER OF SPACIAL NODES (ODD NUMBER)                    EJL00800
C     ND      - DOUBLE PRECISION VALUE OF N                             EUL00810
C     PARRAY  - ARRAY OF PRESSURES FOR PLOTTING                         EUL00820
C     PRI     - INITIAL PRESSURE RATIO ACROSS THE SHOCK                 EUL00830
C     PLTCNT  - COUNTER FOR GRAPHICS ROUTINES                           EUL00840
C     Q       - ABSOLUTE FLUID VELOCITY                                 EUL00850
C     QARRAY  - ARRAY OF VELOCITIES FOR PLOTTING                        EUL00860
C     QLI     - INITIAL VELOCITY LEFT OF THE DIAPHRAGM                  EUL00870
C     QRI     - INITIAL VELOCITY RIGHT OF THE DIAPHRAGM                 EJL00880
C     QQ      - Q+A*S (EXTENDED RIEMANN VARIABLE)                       EUL00890
C     RR      - Q-A*S (EXTENDED RIEMANN VARIABLE)                       EUL00900
C     S       - ENTROPY                                                 EJL00910
C     SARRAY  - ARRAY OF ENTROPY FOR PLOTTING                           EUL00920
C     SIGMA   - SPATIAL LOCATION OF DISCONTINUITIES                     EUL00930
C               SIGMA(L,J) WHERE L INCICATES THE TYPE OF                EUL00940
C                 DISCONTINUITY  AND J INDICATES THE                    EJL00950
C                 TIME LEVEL;  1 - CURRENT LEVEL                        EJL00960
```

```
C                2 - LEVEL BEING CALCULATED                                    EUL00970
C       SKIP    - VARIABLE WHICH INDICATES HOW MANY TIME                       EUL00980
C       T       - TIME SINCE INITIAL CONDITIONS                                EUL00990
C       TRI     - INITIAL TEMP. RATIO ACROSS THE SHOCK                         EUL01000
C       VHEAD   - VELOCITY OF THE HEAD OF THE EXPANSION                        EUL01010
C                 WAVE FOR THE EXACT SOLUTION.                                 EUL01020
C       VTAIL   - VELOCITY OF THE TAIL OF THE EXPANSION                        EUL01030
C                 WAVE FOR THE EXACT SOLUTION.                                 EUL01040
C       VCDE    - VELOCTIY OF THE CONTACT DISCONTINUITY                        EUL01050
C                 FOR THE EXACT SOLUTION.                                      EUL01060
C       VSE     - VELOCITY OF THE SHOCK FOR THE EXACT                          EUL01070
C                 SOLUTION.                                                    EUL01080
C       W       - MACH NO. RELATIVE TO A STANDING SHOCK                        EUL01090
C       XARRAY  - ARRAY OF SPATIAL POSITIONS FOR PLOTTING                      EUL01100
C       XEXACT  - ARRAY OF SIX X VALUES FOR THE EXACT SOLUTION.                EUL01110
C       XINIT   - INITIAL POSITION OF DISCONTINUITY FOR                        EUL01120
C                 EXACT SOLUTION PLOTTING.                                     EUL01130
C       X2      - LOCATION OF NODE UPSTREAM OF DISCONT.                        EUL01140
C                 ALONG THE SPACIAL AXIS.                                      EUL01150
C                 STEPS BETWEEN PLOTS OF THE SOLUTION.                         EUL01160
C       Y       - (N+1)/2                                                      EUL01170
C       YEXACT  - ARRAY OF SIX DENSITY VALUES FOR THE EXACT SOLUTION.          EUL01180
C                                                                              EUL01190
C       *** OTHER VARIABLES ARE DEFINED IN THE                                 EUL01200
C                 SUBROUTINES WHERE THEY ARE USED ****                         EUL01210
C                                                                              EUL01220
C       ++++++++++++++++++++++++++++++++++++++++++++++++                       EUL01230
C       +                                              +                       EUL01240
C       +              PROBLEM  SET - UP               +                       EUL01250
C       +                                              +                       EUL01260
C       ++++++++++++++++++++++++++++++++++++++++++++++++                       EUL01270
C                                                                              EUL01280
C       THE PARTICULAR PROBLEM FOR THIS VERSION IS:                            EUL01290
C                                                                              EUL01300
C           SHOCK TUBE, SINGLE CENTERED DIAPHRAGM WITH                         EUL01310
C           HIGH PRESSURE SIDE TO THE LEFT.                                    EUL01320
C                                                                              EUL01330
C           BOUNDARY CONDITIONS - NONE (INFINITE LENGTH)                       EUL01340
C                                                                              EUL01350
C       ---------  VARIABLE  DECLARATIONS  -------------                       EUL01360
C                                                                              EJL01370
      INTEGER I,J,K,L,N,JSTOP,Y,M,GRAPHS,COUNT,PLTCNT                          EJL01380
      INTEGER SKIP                                                             EUL01390
      REAL*8 TRI,PRI,QLI,QRI,DRI,G,G1,G2,SIGMA(4,2),EE                         EUL01400
      REAL*8 DELT,H,ND,I2(4),X2(4),AR,W,DQ,VS,T                                EJL01410
      REAL VTAIL,VCDE,VSE,DLSH,DLCD,XINIT,VHEAD                                EUL01420
      REAL XEXACT(6),YEXACT(6)                                                 EUL01430
C                                                                              EUL01440
```

```
C      +++++++  USER  INPUT  REQUIRED  HERE  ++++++++++                  EJL01450
C                                                                        EUL01460
C      -------  SET THE DIMENSIONS EQUAL TO N.--------                   EUL01470
       REAL*8 A(51),Q(51),QQ(51),RR(51),S(51)                            EUL01480
       REAL*8 NEWRR(51),NEWS(51),NEWQQ(51)                               EUL01490
       REAL PARRAY(51),DARRAY(51),SARRAY(51)                             EUL01500
       REAL QARRAY(51),XARRAY(51)                                        EUL01510
C                                                                        EUL01520
C      -----  ENTER THE APPROPRIATE VALUES BELOW ------                 EUL01530
       N=51                                                              EUL01540
       M=1                                                               EUL01550
       GRAPHS=2                                                          EJL01560
       SKIP=24                                                           EUL01570
       JSTOP=26                                                          EUL01580
       TRI=1.0D00                                                        EUL01590
       PRI=5.0D00                                                        EUL01600
       DRI=5.0D00                                                        EUL01610
       QLI=0.0D00                                                        EUL01620
       QRI=0.0D00                                                        EUL01630
       G=1.4D00                                                          EUL01640
C                                                                        EUL01650
C      ---------  EXACT  SOLUTION  VALUES  ----------                    EUL01660
       XINIT=0.5                                                         EUL01670
       VHEAD=-1.0                                                        EUL01680
       VTAIL=-0.310557                                                   EUL01690
       VCDE=.574487                                                      EUL01700
       VSE=1.402346                                                      EUL01710
       DLCD=2.713115                                                     EUL01720
       DLSH=1.69344                                                      EUL01730
C                                                                        EUL01740
C      +++++++++  END  OF  USER  INPUT  AREA  ++++++++                   EUL01750
C                                                                        EUL01760
       SIGMA(1,2)=0.500000001D00                                         EUL01770
       SIGMA(2,2)=0.500000001D00                                         EJL01780
       SIGMA(3,2)=0.500000001D00                                         EUL01790
       SIGMA(4,2)=0.500000001D00                                         EJL01800
       EE=0.1D-8                                                         EJL01810
       T=0.0D00                                                          EUL01820
       DO 10 I=1,4                                                       EJL01830
          I2(I)=0.0D00                                                   EJL01840
          X2(I)=0.0D00                                                   EUL01850
          SIGMA(I,1)=0.0D00                                              EUL01860
    10 CONTINUE                                                          EUL01870
       ND=N                                                              EUL01880
       H=1/(ND-1)                                                        EUL01890
       DO 11 I=1,N                                                       EUL01900
          A(I)=0.0D00                                                    EUL01910
          Q(I)=0.0D00                                                    EUL01920
```

```
            NEWQQ(I)=0.0D00                                                 EUL01930
            NEWRR(I)=0.0D00                                                 EUL01940
            NEWS(I)=0.0D00                                                  EUL01950
            QARRAY(I)=0.0                                                   EUL01960
            PARRAY(I)=0.0                                                   EUL01970
            DARRAY(I)=0.0                                                   EUL01980
            SARRAY(I)=0.0                                                   EUL01990
            XARRAY(I)=(I-1)*SNGL(H)                                         EUL02000
         11 CONTINUE                                                        EUL02010
            DELT=2.0D00                                                     EUL02020
            Y=(N+1)/2                                                       EUL02030
            AR=1.0D00                                                       EUL02040
            DQ=0.0D00                                                       EUL02050
            W=1.0D00                                                        EUL02060
            VS=0.0D00                                                       EUL02070
            G1=1/(G-1)                                                      EUL02080
            G2=2/(G-1)                                                      EUL02090
            DO 12 I=1,Y                                                     EUL02100
               S(I)=G2-(G1/G)*DLOG(PRI/((DRI)**G))                          EUL02110
               QQ(I)=QLI+DSQRT(TRI)*S(I)                                    EUL02120
               RR(I)=QLI-DSQRT(TRI)*S(I)                                    EUL02130
         12 CONTINUE                                                        EUL02140
            Y=(N+3)/2                                                       EUL02150
            DO 13 I=Y,N                                                     EUL02160
               S(I)=G2                                                      EUL02170
               QQ(I)=QRI+S(I)                                               EUL02180
               RR(I)=QRI-S(I)                                               EUL02190
         13 CONTINUE                                                        EUL02200
C                                                                           EUL02210
            IF (GRAPHS.GT.0) THEN                                           EUL02220
               CALL COMPRS                                                  EUL02230
C              CALL TEK618                                                  EUL02240
               CALL HWROT('AUTO')                                           EUL02250
               CALL HWSCAL('SCREEN')                                        EUL02260
               IF (GRAPHS.EQ.2) THEN                                        EUL02270
                 CALL PAGE(11.0,8.5)                                        EUL02280
               ELSE                                                         EUL02290
                 CALL PAGE(8.5,11.0)                                        EUL02300
               END IF                                                       EUL02310
               IF (GRAPHS.EQ.1) THEN                                        EUL02320
               CALL BORDER(JSTOP)                                           EUL02330
               END IF                                                       EUL02340
            END IF                                                          EUL02350
            J=1                                                             EUL02360
            COUNT=1                                                         EUL02370
            IF (GRAPHS.EQ.1) THEN                                           EUL02380
               CALL PLOT(J,JSTOP,N,QQ,RR,S,H,XARRAY,PARRAY,                 EUL02390
           #DARRAY,QARRAY,SARRAY,G,G1,G2)                                   EUL02400
```

```
      END IF                                                                    EUL02410
   14 IF (J.EQ.JSTOP) GOTO 15                                                   EJL02420
      PLTCNT=J/SKIP                                                             EUL02430
C                                                                               EJL02440
      CALL TIME(N,QQ,RR,S,DELT,H)                                               EUL02450
C                                                                               EJL02460
      CALL TRAK(N,SIGMA,H,QQ,RR,S,G,G1,G2,M,DELT,I2,X2,W,AR,DQ,VS,J)            EUL02470
C                                                                               EJL02480
      CALL SWEEP(N,H,SIGMA,QQ,RR,S,DELT,EE,Q,A,NEWQQ,NEWRR,NEWS,I2,G2,J)        EUL02490
C                                                                               EJL02500
      CALL JUMP(N,SIGMA,H,QQ,RR,S,G,G1,M,I2,X2,W,AR,DQ,VS,J)                    EUL02510
C                                                                               EUL02520
      T=T+DELT                                                                  EUL02530
C                                                                               EUL02540
      IF ((COUNT.EQ.PLTCNT*SKIP).AND.(GRAPHS.EQ.1)) THEN                        EUL02550
         CALL PLOT(J,JSTOP,N,QQ,RR,S,H,XARRAY,PARRAY,                           EUL02560
     #DARRAY,QARRAY,SARRAY,G,G1,G2)                                             EUL02570
      END IF                                                                    EJL02580
      IF ((COUNT.EQ.PLTCNT*SKIP).AND.(GRAPHS.EQ.0)) THEN                        EUL02590
         CALL LIST(N,SIGMA,QQ,RR,S,G,G1,G2,J,T,DELT,VS)                         EUL02600
      END IF                                                                    EUL02610
      IF ((COUNT.EQ.PLTCNT*SKIP).AND.(GRAPHS.EQ.2)) THEN                        EUL02620
         CALL EXACT(N,XINIT,T,VHEAD,VTAIL,VCDE,VSE,DLCD,DLSH,QQ,RR,S,H,         EUL02630
     #XARRAY,DARRAY,G,G1,G2,DRI)                                                EJL02640
      END IF                                                                    EUL02650
C                                                                               EJL02660
      J=J+1                                                                     EUL02670
      COUNT=COUNT+1                                                             EUL02680
      GO TO 14                                                                  EUL02690
   15 CALL DONEPL                                                               EUL02700
      END                                                                       EUL02710
C     ******************************************                                EUL02720
C     ******************************************                                EUL02730
C     ******************************************                                EUL02740
C                                                                               EUL02750
      SUBROUTINE LIST(N,SIGMA,QQ,RR,S,G,G1,G2,J,T,DELT,VS)                      EUL02760
C                                                                               EJL02770
C     ++++++++++++++++++++++++++++++++++++++++++++++++                          EUL02780
C     +                                              +                          EJL02790
C     +       TABULAR  RESULTS  SUBROUTINE           +                          EUL02800
C     +                                              +                          EUL02810
C     ++++++++++++++++++++++++++++++++++++++++++++++++                          EUL02820
C                                                                               EUL02830
C     ---------- VARIABLE DEFINITIONS --------------                           EUL02840
C                                                                               EUL02850
C     DENS  - DENSITY                                                           EUL02860
C     PRESS - PRESSURE                                                          EUL02870
C     TEMP  - TEMPERATURE                                                       EJL02880
```

```
C                                                                       EJL02890
      INTEGER I,J,N,L                                                   EUL02900
      REAL*8 SIGMA(4,2),QQ(N),RR(N),S(N),PRESS                          EUL02910
      REAL*8 TEMP,DENS,G,G1,G2,Q,T,DELT,VS                              EUL02920
C                                                                       EUL02930
      WRITE(9,*) 'TIME LEVEL',J,'                ELAPSED TIME IS',T     EUL02940
      WRITE(9,*) 'TIME STEP IS',DELT,'    SHOCK VELOCITY IS',VS         EUL02950
      WRITE(9,*) ' '                                                    EJL02960
      WRITE(9,*) '          NODE          VELOCITY                DENSITY   EUL02970
     # PRESSURE'                                                        EUL02980
      WRITE(9,*) ' '                                                    EJL02990
      DO 61 I=1,N                                                       EUL03000
      TEMP=(QQ(I)-RR(I))*(QQ(I)-RR(I))/(4*S(I)*S(I))                    EJL03010
      DENS=((1/TEMP)*DEXP(G*(1-G)*(S(I)-G2)))**(-G1)                    EUL03020
      PRESS=TEMP*DENS                                                   EUL03030
      Q=(QQ(I)+RR(I))/2.0D00                                            EUL03040
         WRITE(9,*) I,Q,DENS,PRESS                                      EUL03050
   61 CONTINUE                                                          EUL03060
      WRITE(9,*) ' '                                                    EJL03070
      WRITE(9,*) ' '                                                    EUL03080
      WRITE(9,*) '            NODE          QQ                   RR   EUL03090
     # MODIFIED S'                                                      EJL03100
      WRITE(9,*) ' '                                                    EUL03110
      DO 62 I=1,N                                                       EJL03120
         WRITE(9,*) I,QQ(I),RR(I),S(I)                                  EUL03130
   62 CONTINUE                                                          EUL03140
      WRITE(9,*) ' '                                                    EUL03150
      WRITE(9,*) ' '                                                    EJL03160
      WRITE(9,*) '      DISCONTINUITY LOCATIONS AT TIME LEVEL',J        EUL03170
      WRITE(9,*) ' '                                                    EUL03180
      WRITE(9,*) '        TYPE          LOCATION'                       EUL03190
      DO 63 L=1,4                                                       EUL03200
         WRITE(9,*) L,'     ',SIGMA(L,2)                               EUL03210
   63 CONTINUE                                                          EUL03220
      WRITE(9,*) ' '                                                    EJL03230
      WRITE(9,*) '-------------------------------------------------------- EJL03240
     #-------------------------------------------------------------'     EJL03250
      WRITE(9,*) ' '                                                    EUL03260
      WRITE(9,*) ' '                                                    EUL03270
      RETURN                                                            EUL03280
      END                                                               EUL03290
C                                                                       EJL03300
C                                                                       EUL03310
C                                                                       EUL03320
C                                                                       EUL03330
C                                                                       EUL03340
C                                                                       EUL03350
      SUBROUTINE TIME(N,QQ,RR,S,DELT,H)                                 EUL03360
```

```
C                                                                       EUL03370
CCC   ********************************************                      EUL03380
CCC   *                                          *                      EUL03390
CCC   *      CALCULATE  TIME  STEP  SUBROUTINE   *                      EUL03400
CCC   *                                          *                      EUL03410
CCC   ********************************************                      EUL03420
CCC                                                                     EJL03430
CCC   ---------- VARIABLE  DEFINITIONS  ----------                      EUL03440
CCC                                                                     EJL03450
CCC   TMIN - RUNNING VALUE OF THE MINIMUM TIME STEP                     EUL03460
C                                                                       EUL03470
      INTEGER N,I                                                       EUL03480
      REAL*8 H,A,QQ(N),RR(N),S(N),DELT,TMIN,Q                           EJL03490
      TMIN=2.0D00                                                       EUL03500
      DO 21 I=1,N                                                       EJL03510
         A=(QQ(I)-RR(I))/(2*S(I))                                       EUL03520
         Q=(QQ(I)+RR(I))/2.0D00                                         EUL03530
         DELT=H/(DABS(Q+A))                                             EUL03540
         IF (DELT.LT.TMIN) THEN                                         EUL03550
         TMIN=DELT                                                      EUL03560
         END IF                                                         EUL03570
   21 CONTINUE                                                          EUL03580
      DELT=0.99D00*TMIN                                                 EJL03590
      RETURN                                                            EUL03600
      END                                                               EUL03610
C                                                                       EUL03620
      SUBROUTINE TRAK(N,SIGMA,H,QQ,RR,S,G,G1,G2,M,DELT,I2,X2,W,AR,DQ,VS,EUL03630
     #J)                                                                EJL03640
CCC   ********************************************                      EUL03650
CCC   *                                          *                      EUL03660
CCC   *      DISCONTINUITY TRACKING SUBROUTINE   *                      EUL03670
CCC   *                                          *                      EJL03680
CCC   ********************************************                      EUL03690
CCC                                                                     EJL03700
CCC   ----------  VARIABLE  DEFINITIONS ----------                      EUL03710
CCC                                                                     EJL03720
CCC   AR   - THE RATIO OF SOUND SPEED ACROSS A                          EUL03730
CCC          SHOCK, A/B                                                 EJL03740
CCC   DR   - THE RATIO OF THE DENSITY ACROSS A                          EUL03750
CCC          SHOCK, B/A                                                 EJL03760
CCC   DQ   - THE JUMP IN VELOCITY ACROSS THE SHOCK                      EUL03770
CCC          DIVIDED BY THE SOUND SPEED AT B.                           EUL03780
CCC   I2   - NUMBER OF THE NODE TO THE RIGHT OF A                       EUL03790
CCC          DISCONTINUITY.                                             EUL03800
CCC   PR   - THE RATIO OF THE PRESSURE ACROSS A                         EUL03810
CCC          SHOCK, A/B                                                 EUL03820
CCC   QQJO - THE MEASURED JUMP IN QQ ACROSS THE SHOCK,                  EUL03830
C            FROM A TO B.                                               EUL03840
```

```
C      QQJE - THE JUMP IN QQ ACROSS THE SHOCK CALCULATED ANALYTICALLY     EUL03850
C             AS A FUNCTION OF W.                                         EUL03860
C      VS   - THE SHOCK SPEED                                             EUL03870
C      X2   - LOCATION OF THE NODE TO THE RIGHT OF A                      EUL03880
C             DISCONTINUITY.                                              EJL03890
C      *A   - DENOTES THE VALUE OF A VARIABLE AT THE NODE                 EUL03900
C             TO THE LEFT OF A DISCONTINUITY.  * CAN                      EUL03910
C             BE ANY VARIABLE NAME.                                       EUL03920
C      *B   - DENOTES THE VALUE OF A VARIABLE AT THE NODE                 EJL03930
C             TO THE RIGHT OF A DISCONTINUITY.                            EUL03940
C                                                                         EJL03950
      INTEGER N,I,Y,I2(4),L,M,J                                           EUL03960
      REAL*8 SIGMA(4,2),X2(4),X,H,XA(4),XB(4),AB,SA,SB                    EJL03970
      REAL*8 QQA,QQB,RRA,RRB,RR(N),QQ(N),S(N),QQJE,TS                     EUL03980
      REAL*8 QQJ0,QQJ1,W,DQ,AR,PR,DR,G,G1,G2,VS,DELT                      EUL03990
C                                                                         EUL04000
CC     +++++++  LOCATING THE UPSTREAM NODE ++++++++++                     EUL04010
C                                                                         EJL04020
      DO 31 L=1,M                                                         EUL04030
         SIGMA(L,1)=SIGMA(L,2)                                            EUL04040
         Y=0                                                              EUL04050
         X=H                                                              EJL04060
         I=2                                                              EUL04070
   30    IF (.NOT.(Y.EQ.0)) GOTO 31                                       EUL04080
           IF (SIGMA(L,1).LE.X) THEN                                      EUL04090
              X2(L)=X                                                     EUL04100
              I2(L)=I                                                     EUL04110
              Y=1                                                         EJL04120
           END IF                                                         EUL04130
         X=X+H                                                            EUL04140
         I=I+1                                                            EJL04150
         GOTO 30                                                          EUL04160
   31   CONTINUE                                                         EUL04170
C                                                                         EUL04180
CC     ++++++++++++  DETERMINING SHOCK SPEED  ++++++++                    EUL04190
C                                                                         EUL04200
      XA(1)=(SIGMA(1,1)+H-X2(1))/H                                        EUL04210
      XB(1)=(X2(1)-SIGMA(1,1))/H                                          EUL04220
      RRA=RR(I2(1)-1)                                                     EUL04230
      RRB=RR(I2(1))                                                       EUL04240
      QQA=QQ(I2(1)-1)                                                     EUL04250
      QQB=QQ(I2(1))                                                       EUL04260
      SA=S(I2(1)-1)                                                       EUL04270
      SB=S(I2(1))                                                         EJL04280
      AB=(QQB-RRB)/(2*SB)                                                 EUL04290
      AA=(QQA-RRA)/(2*SA)                                                 EUL04300
      QQJ0=(QQA-QQB)/AB                                                   EUL04310
      QQJ1=QQJ0                                                           EJL04320
```

```
  300  W=5.513294D00-DSQRT(3.0396408D01-((QQJ1+2.7574D00)/0.286337D00))   EUL04330
       DQ=2*(W*W-1)/(W*(G+1))                                             EUL04340
       AR=DSQRT(2*(G-1)*(1+((G-1)*W*W/2))*(G*G2*W*W-1))/((G+1)*W)         EUL04350
       PR=(2*G/(G+1))*W*W-((G-1)/(G+1))                                   EUL04360
       DR=((G-1)*W*W+2)/((G+1)*W*W)                                       EUL04370
       QQJE=DQ+(AR-1)*G2-(AR*G1/G)*DLOG(PR*(DR**G))                       EUL04380
C      WRITE(9,*) W,QQJE,QQJ0                                             EUL04390
       IF (DABS(QQJE-QQJ0).LT.0.1D-5) GOTO 310                            EUL04400
       QQJ1=(QQJ0-QQJE)+QQJ1                                              EUL04410
       GOTO 300                                                           EUL04420
  310  VS=(QQB+RRB)+W*AB                                                  EUL04430
       TS=H/VS                                                            EUL04440
  311  IF (TS.LT.DELT) THEN                                               EUL04450
          DELT=0.99D00*TS                                                 EUL04460
       END IF                                                             EUL04470
       SIGMA(1,2)=VS*DELT+SIGMA(1,1)                                      EUL04480
       RETURN                                                             EUL04490
       END                                                                EUL04500
C                                                                         EUL04510
       SUBROUTINE SWEEP(N,H,SIGMA,QQ,RR,S,DELT,EE,Q,A,NEWQQ,NEWRR,NEWS,   EUL04520
      #I2,G2,J)                                                           EUL04530
C                                                                         EUL04540
CC     ******************************************                         EUL04550
CC     *                                        *                         EUL04560
CC     *         SPACE  SWEEPING  SUBROUTINE    *                         EUL04570
CC     *                                        *                         EUL04580
CC     ******************************************                         EUL04590
CC                                                                        EUL04600
CC     ------------ VARIABLE  DEFINITIONS -----------                     EUL04610
CC                                                                        EUL04620
CC     DELQQH - CHANGE IN QQ FROM I TO I+1                                EUL04630
CC     DELQQL - CHANGE IN QQ FROM I-1 TO I                                EUL04640
CC     DELRRH - CHANGE IN RR FROM I TO I+1                                EUL04650
CC     DELRRL - CHANGE IN RR FROM I-1 TO I                                EUL04660
CC     DELSH  - CHANGE IN S FROM I TO I+1                                 EUL04670
CC     DELSL  - CHANGE IN S FROM I-1 TO I                                 EUL04680
CC     DELAH  - CHANGE IN A FROM I TO I+1                                 EUL04690
CC     DELAL  - CHANGE IN A FROM I-1 TO I                                 EUL04700
CC     DELQH  - CHANGE IN Q FROM I TP I+1                                 EUL04710
CC     DELQL  - CHANGE IN Q FROM I-1 TO I                                 EUL04720
CC     DELX   - INTERPOLATION DISTANCE (LMD*DELT)                         EUL04730
CC     DLTA** - PREFIX WHICH INDICATES THE SPATIAL                        EUL04740
CC              CHANGE IN ** FOR ONE TIME STEP.                           EUL04750
C      E(K)   - ACTUAL ERROR IN CHARACTERISTIC SLOPE  CALCULATION.        EUL04760
C    INTEG(K)- RESULT OF INTEGRATING Z(K).                                EUL04770
C     LMD    - SLOPE OF THE CHARACTERISTICS (Q+A,Q,Q-A)                   EUL04780
C    NEW**(I)- STORED VALUES OF ** FOR THE NEXT TIME LVL                  EUL04790
C    *PRIM(K)- SUFFIX WHICH INDICATES THE SPATIAL                         EUL04800
```

```
C                DERIVITIVE OF * AT THE CURRENT TIME LEVEL.                    EUL04810
C       QQINT    - VALUE OF ** INTERPOLATED BETWEEN NODES                      EUL04820
C                  ON THE CURRENT TIME LEVEL.                                  EUL04830
C       **STEP   - THE CHANGE IN TIME OF ** AT A NODE                          EUL04840
C                  USED TO STEP UP TO THE NEXT TIME LEVEL                      EUL04850
C       X        - LOCATION IN SPACIAL PLANE (I-1)*H                           EUL04860
C       Z(K)     - RIGHT SIDE OF THE K'TH EQUATION.                            EUL04870
C                                                                              EUL04880
      INTEGER N,K,I,I2(4),J                                                    EUL04890
      REAL*8 X,H,SIGMA(4,2),QQ(N),RR(N),S(N),Q(N),A(N)                         EUL04900
      REAL*8 DELQQH,DELQQL,DELRRH,DELRRL,DELSH,DELSL                           EUL04910
      REAL*8 DELAH,DELAL,DELQH,DELQL,LMD(3),DELX(3),DELT                       EUL04920
      REAL*8 QINT(3),AINT(3),QQINT,RRINT,SINT,EE,E(3)                          EUL04930
      REAL*8 NEWQQ(N),NEWRR(N),NEWS(N),QQSTEP                                  EUL04940
      REAL*8 RRSTEP,SSTEP,INTEG(3),Z(3),DLTAQQ                                 EUL04950
      REAL*8 DLTARR,DLTAS,APRIM(3),QPRIM(3)                                    EUL04960
      REAL*8 G2,ABAR(3)                                                        EUL04970
      DO 40 I=1,N                                                              EUL04980
         Q(I)=(QQ(I)+RR(I))/2.0D00                                             EUL04990
         A(I)=((QQ(I)-RR(I))/(2*S(I)))                                         EUL05000
   40 CONTINUE                                                                 EUL05010
      I=2                                                                      EUL05020
   41 IF (I.EQ.N) GOTO 49                                                      EUL05030
        X=(I-1)*H                                                              EUL05040
        DELQQH=QQ(I+1)-QQ(I)                                                   EUL05050
        DELQQL=QQ(I)-QQ(I-1)                                                   EUL05060
        DELRRH=RR(I+1)-RR(I)                                                   EUL05070
        DELRRL=RR(I)-RR(I-1)                                                   EUL05080
        DELSH=S(I+1)-S(I)                                                      EUL05090
        DELSL=S(I)-S(I-1)                                                      EUL05100
        DELAH=A(I+1)-A(I)                                                      EUL05110
        DELAL=A(I)-A(I-1)                                                      EUL05120
        DELQH=Q(I+1)-Q(I)                                                      EUL05130
        DELQL=Q(I)-Q(I-1)                                                      EUL05140
C                                                                              EUL05150
C       -----  TEST FOR EXISTENCE OF A DISCONTINUITY                           EUL05160
C                                                                              EUL05170
        IF ((SIGMA(1,1).GT.X-H).AND.(SIGMA(1,1).LT.X)) THEN                    EUL05180
           IF (SIGMA(1,2).GT.X) THEN                                           EUL05190
           QQSTEP=0.0D00                                                       EUL05200
           RRSTEP=0.0D00                                                       EUL05210
           SSTEP=0.0D00                                                        EUL05220
           GOTO 48                                                             EUL05230
           END IF                                                              EUL05240
           CALL LSHOCK(Q,A,RR,QQ,S,N,DELQQH,DELRRH,DELSH,DELAH                 EUL05250
     #,DELQH,H,EE,I,DELT,AINT,QQINT,RRINT,SINT)                                EUL05260
           GOTO 47                                                             EUL05270
        END IF                                                                 EUL05280
```

```
C                                                                       EUL05290
      IF ((SIGMA(1,1).GT.X).AND.(SIGMA(1,1).LT.X+H).AND.                EUL05300
     #(Q(I).LE.A(I))) THEN                                              EUL05310
        CALL RSHOCK(Q,A,RR,QQ,S,N,DELQQL,DELRRL,DELSL,DELQL,DELAL       EUL05320
     #,H,EE,I,DELT,AINT,QQINT,RRINT,SINT)                               EUL05330
        GOTO 47                                                         EUL05340
      END IF                                                            EUL05350
C                                                                       EUL05360
      LMD(1)=QQ(I-1)+A(I-1)                                             EUL05370
      LMD(2)=Q(I)                                                       EUL05380
      IF (Q(I).LT.A(I)) GOTO 42                                         EUL05390
      LMD(3)=Q(I)-A(I)                                                  EUL05400
      GOTO 43                                                           EUL05410
42    LMD(3)=Q(I+1)-A(I+1)                                              EUL05420
43    K=1                                                               EUL05430
50    IF (K.EQ.4) GOTO 51                                               EUL05440
        DELX(K)=DELT*LMD(K)                                             EUL05450
        IF (LMD(K).GT.0.0D00) GOTO 44                                   EUL05460
        IF (LMD(K).LT.0.0D00) GOTO 45                                   EUL05470
        QINT(K)=Q(I)                                                    EUL05480
        AINT(K)=A(I)                                                    EUL05490
        GOTO 46                                                         EUL05500
44      QINT(K)=Q(I)-DELX(K)*DELQL/H                                    EUL05510
        AINT(K)=A(I)-DELX(K)*DELAL/H                                    EUL05520
        GOTO 46                                                         EUL05530
45      QINT(K)=Q(I)-DELX(K)*DELQH/H                                    EUL05540
        AINT(K)=A(I)-DELX(K)*DELAH/H                                    EUL05550
46      K=K+1                                                           EUL05560
      GOTO 50                                                           EUL05570
51    CONTINUE                                                          EUL05580
C                                                                       EUL05590
      E(1)=DABS(LMD(1)-(QINT(1)+AINT(1)))                               EUL05600
      E(2)=DABS(LMD(2)-QINT(2))                                         EUL05610
      E(3)=DABS(LMD(3)-(QINT(3)-AINT(3)))                               EUL05620
      LMD(1)=QINT(1)+AINT(1)                                            EUL05630
      LMD(2)=QINT(2)                                                    EUL05640
      LMD(3)=QINT(3)-AINT(3)                                            EUL05650
      IF ((E(1).GT.EE).OR.(E(2).GT.EE).OR.(E(3).GT.EE)) GOTO 43         EUL05660
C                                                                       EUL05670
C   END OF ITERATION FOR  LMD(K) & DELX(K) FOR THIS NODE                EUL05680
C                                                                       EUL05690
      QQINT=QQ(I)-DELX(1)*DELQQL/H                                      EUL05700
      IF (LMD(2).LE.0.0D00) THEN                                        EUL05710
        SINT=S(I)-DELX(2)*DELSH/H                                       EUL05720
      ELSE                                                              EUL05730
        SINT=S(I)-DELX(2)*DELSL/H                                       EUL05740
      END IF                                                            EUL05750
      IF (LMD(3).LE.0.0D00) THEN                                        EUL05760
```

```
            RRINT=RR(I)-DELX(3)*DELRRH/H                                    EUL05770
         ELSE                                                               EJL05780
            RRINT=RR(I)-DELX(3)*DELRRL/H                                    EUL05790
         END IF                                                             EJL05800
C                                                                           EUL05810
C     --------  CALCULATE DLTA QQ, DLTA RR & DLTA S                         EUL05820
C                                                                           EUL05830
   47    DLTAQQ=QQINT-QQ(I)                                                 EUL05840
         DLTARR=RRINT-RR(I)                                                 EUL05850
         DLTAS=SINT-S(I)                                                    EUL05860
C                                                                           EJL05870
C        --------  CALCULATE SPATIAL DERIVITIVES -----                      EUL05880
C                                                                           EUL05890
         IF ((I.EQ.I2(1)-1).OR.(Q(I).GE.A(I))) THEN                         EUL05900
            QPRIM(3)=(Q(I)-Q(I-1))/H                                        EJL05910
            APRIM(3)=(A(I)-A(I-1))/H                                        EUL05920
         ELSE                                                               EJL05930
            QPRIM(3)=(Q(I+1)-Q(I))/H                                        EUL05940
            APRIM(3)=(A(I+1)-A(I))/H                                        EUL05950
         END IF                                                             EUL05960
         IF (I.EQ.I2(1)) THEN                                               EUL05970
            QPRIM(1)=(Q(I+1)-Q(I))/H                                        EUL05980
            APRIM(1)=(A(I+1)-A(I))/H                                        EUL05990
         ELSE                                                               EJL06000
            QPRIM(1)=(Q(I)-Q(I-1))/H                                        EUL06010
            APRIM(1)=(A(I)-A(I-1))/H                                        EUL06020
         END IF                                                             EUL06030
         QPRIM(2)=0.0D00                                                    EUL06040
C                                                                           EJL06050
C        ------  CALCULATE  Z(K)'S ----------                               EUL06060
C                                                                           EUL06070
C                                                                           EJL06080
         ABAR(1)=(AINT(1)+A(I))/2.0D00                                      EUL06090
         ABAR(3)=(AINT(3)+A(I))/2.0D00                                      EUL06100
         ABAR(2)=0.0D00                                                     EJL06110
         Z(1)=-(1.0D00/G2)*ABAR(1)*(S(I)-G2)*(QPRIM(1)-G2*APRIM(1))         EUL06120
         Z(3)=(1.0D00/G2)*ABAR(3)*(S(I)-G2)*(QPRIM(3)+G2*APRIM(3))          EJL06130
         Z(2)=0.0D00                                                        EUL06140
C                                                                           EUL06150
C        ----- INTEGRATE THE Z(K)'S ------------                           EJL06160
C                                                                           EUL06170
         INTEG(2)=0.0D00                                                    EUL06180
         INTEG(1)=Z(1)*DELT                                                 EUL06190
         INTEG(3)=Z(3)*DELT                                                 EUL06200
C                                                                           EJL06210
C        --------  SOLVE THE EQUATION -----------                           EUL06220
C                                                                           EUL06230
         QQSTEP=DLTAQQ+INTEG(1)                                             EJL06240
```

```
      RRSTEP=DLTARR+INTEG(3)                                            EUL06250
      SSTEP=DLTAS+INTEG(2)                                              EUL06260
C                                                                       EUL06270
C     -------- STORE THE SOLUTION -----------                           EUL06280
C                                                                       EUL06290
 48   NEWQQ(I)=QQ(I)+QQSTEP                                             EUL06300
      NEWRR(I)=RR(I)+RRSTEP                                             EUL06310
      NEWS(I)=S(I)+SSTEP                                                EUL06320
C                                                                       EUL06330
C     --------- GO TO NEXT NODE  ------------                           EUL06340
C                                                                       EUL06350
      I=I+1                                                             EUL06360
      GOTO 41                                                           EUL06370
 49   CONTINUE                                                          EUL06380
C                                                                       EUL06390
C     --------- UPDATE THE VARIABLES  ----------                       EUL06400
C                                                                       EUL06410
      I=2                                                               EUL06420
 52   IF (I.EQ.N) GOTO 53                                               EUL06430
        RR(I)=NEWRR(I)                                                  EUL06440
        QQ(I)=NEWQQ(I)                                                  EUL06450
        S(I)=NEWS(I)                                                    EUL06460
      I=I+1                                                             EUL06470
      GOTO 52                                                           EUL06480
 53   CONTINUE                                                          EUL06490
      RETURN                                                            EUL06500
      END                                                               EUL06510
C                                                                       EUL06520
      SUBROUTINE LSHOCK(Q,A,RR,QQ,S,N,DELQQH,DELRRH,DELSH,DELAH         EUL06530
     #,DELQH,H,EE,I,DELT,AINT,QQINT,RRINT,SINT)                         EUL06540
C                                                                       EUL06550
C     ***************************************************               EUL06560
C     *                                                 *               EUL06570
C     *            LSHOCK  SUBROUTINE                   *               EUL06580
C     *                                                 *               EUL06590
C     ***************************************************               EUL06600
C                                                                       EUL06610
      INTEGER I,N,K                                                     EUL06620
      REAL*8 Q(N),A(N),RR(N),QQ(N),S(N),DELQQH,DELRRH                   EUL06630
      REAL*8 DELSH,DELAH,DELQH,LMD(3),QINT(3),AINT(3),H                 EUL06640
      REAL*8 DELX(3),QQINT,RRINT,SINT,E(3),DELT,EE                      EUL06650
C                                                                       EUL06660
      LMD(1)=Q(I)+A(I)                                                  EUL06670
      LMD(2)=Q(I)                                                       EUL06680
      LMD(3)=Q(I+1)-A(I+1)                                              EUL06690
 410  DO 411 K=1,3                                                      EUL06700
        DELX(K)=DELT*LMD(K)                                             EUL06710
        QINT(K)=Q(I)-DELX(K)*DELQH/H                                    EUL06720
```

```
      AINT(K)=A(I)-DELX(K)*DELAH/H                                      EUL06730
  411 CONTINUE                                                          EJL06740
      E(1)=DABS(LMD(1)-(QINT(1)+AINT(1)))                               EUL06750
      E(2)=DABS(LMD(2)-QINT(2))                                         EJL06760
      E(3)=DABS(LMD(3)-(QINT(3)-AINT(3)))                               EUL06770
      LMD(1)=QINT(1)+AINT(1)                                            EUL06780
      LMD(2)=QINT(2)                                                    EUL06790
      LMD(3)=QINT(3)-AINT(3)                                            EUL06800
      IF ((E(1).GT.EE).OR.(E(2).GT.EE).OR.(E(3).GT.EE)) GOTO 410        EJL06810
C                                                                       EJL06820
      QQINT=QQ(I)-DELX(1)*DELQQH/H                                      EUL06830
      RRINT=RR(I)-DELX(3)*DELRRH/H                                      EUL06840
      SINT=S(I)-DELX(2)*DELSH/H                                         EUL06850
      RETURN                                                            EUL06860
      END                                                               EUL06870
C                                                                       EJL06880
      SUBROUTINE RSHOCK(Q,A,RR,QQ,S,N,DELQQL,DELRRL,DELSL,DELQL,DELAL   EUL06890
     #,H,EE,I,DELT,AINT,QQINT,RRINT,SINT)                               EJL06900
C                                                                       EUL06910
CCC   ************************************************                EUL06920
CCC   *                                              *                EJL06930
CCC   *            RSHOCK   SUBROUTINE               *                EUL06940
CCC   *                                              *                EUL06950
CCC   ************************************************                EUL06960
C                                                                       EJL06970
      INTEGER I,N,K                                                     EUL06980
      REAL*8 Q(N),A(N),RR(N),QQ(N),S(N),DELQQL,DELRRL                   EUL06990
      REAL*8 DELSL,DELQL,DELAL,LMD(3),QINT(3),AINT(3)                   EUL07000
      REAL*8 DELX(3),H,QQINT,RRINT,SINT,EE,E(3)                         EUL07010
      REAL*8 DELT                                                       EUL07020
  419   LMD(1)=Q(I-1)+A(I-1)                                            EUL07030
        LMD(2)=Q(I)                                                     EUL07040
        LMD(3)=Q(I)-A(I)                                                EUL07050
  420   K=1                                                             EUL07060
  425   IF (K.EQ.4) GOTO 465                                            EJL07070
          DELX(K)=DELT*LMD(K)                                           EUL07080
  440     QINT(K)=Q(I)-DELX(K)*DELQL/H                                  EJL07090
          AINT(K)=A(I)-DELX(K)*DELAL/H                                  EUL07100
  460     K=K+1                                                         EUL07110
        GOTO 425                                                        EUL07120
  465   CONTINUE                                                        EUL07130
C                                                                       EUL07140
        E(1)=DABS(LMD(1)-(QINT(1)+AINT(1)))                             EUL07150
        E(2)=DABS(LMD(2)-QINT(2))                                       EJL07160
        E(3)=DABS(LMD(3)-(QINT(3)-AINT(3)))                             EUL07170
        LMD(1)=QINT(1)+AINT(1)                                          EUL07180
        LMD(2)=QINT(2)                                                  EUL07190
        LMD(3)=QINT(3)-AINT(3)                                          EUL07200
```

```
        IF ((E(1).GT.EE).OR.(E(2).GT.EE).OR.(E(3).GT.EE)) GOTO 420      EUL07210
C                                                                       EUL07220
      QQINT=QQ(I)-DELX(1)*DELQQL/H                                      EUL07230
      RRINT=RR(I)-DELX(3)*DELRRL/H                                      EUL07240
      SINT=S(I)-DELX(2)*DELSL/H                                         EUL07250
C                                                                       EJL07260
      RETURN                                                            EUL07270
      END                                                               EUL07280
C                                                                       EUL07290
      SUBROUTINE JUMP(N,SIGMA,H,QQ,RR,S,G,G1,M,I2,X2,W,AR,DQ,VS,J)      EUL07300
C                                                                       EUL07310
C     ************************************************                  EUL07320
C     *                                              *                  EJL07330
C     *    DISCONTINUITY CORRECTION SUBROUTINE       *                  EUL07340
C     *                                              *                  EJL07350
C     ************************************************                  EUL07360
C                                                                       EJL07370
C     U  - THE VELOCITY RELATIVE TO THE STEADY SHOCK WAVE.              EUL07380
C                                                                       EUL07390
      INTEGER N,M,I,Y,L,I2(4),J                                         EUL07400
      REAL*8 SIGMA(4,2),H,QQ(N),RR(N),S(N),X,XA(4)                      EUL07410
      REAL*8 XB(4),AB,SA,SB,QQA,QQB,RRA,RRB,W,DQ,AR,G                   EUL07420
      REAL*8 G1,VS,QA,QB,AA,UA,UB,SA1,SA2,X2(4)                         EJL07430
C     ++++++++  LOCATING THE UPSTREAM NODE ++++++++++                   EUL07440
C                                                                       EUL07450
      DO 62 L=1,M                                                       EUL07460
         Y=0                                                            EUL07470
         X=H                                                            EJL07480
         I=2                                                            EUL07490
   60    IF (.NOT.(Y.EQ.0)) GOTO 62                                     EUL07500
           IF (SIGMA(L,2).LE.X) THEN                                    EUL07510
              X2(L)=X                                                   EUL07520
              I2(L)=I                                                   EUL07530
              Y=1                                                       EJL07540
           END IF                                                       EUL07550
         X=X+H                                                          EUL07560
         I=I+1                                                          EUL07570
         GOTO 60                                                        EUL07580
   62  CONTINUE                                                         EUL07590
C                                                                       EJL07600
C     +++++ CORRECTING THE VALUES AT NODE "A" ++++++++                  EUL07610
C                                                                       EUL07620
      IF ((SIGMA(1,2).GT.X2(1)-H).AND.(SIGMA(1,1).LT.X2(1)-H)) THEN     EUL07630
      XA(1)=(SIGMA(1,2)+H-X2(1))/H                                      EUL07640
      XB(1)=(X2(1)-SIGMA(1,2))/H                                        EJL07650
      RRB=RR(I2(1))-XB(1)*(RR(I2(1)+1)-RR(I2(1)))                       EUL07660
      QQB=QQ(I2(1))-XB(1)*(QQ(I2(1)+1)-QQ(I2(1)))                       EUL07670
      SB=S(I2(1))-XB(1)*(S(I2(1)+1)-S(I2(1)))                           EUL07680
```

```
      AB=(QQB-RRB)/(2*SB)                                                    EUL07690
      AA=AB*AR                                                               EUL07700
      SA1=(G1/G)*DLOG((2*G*W*W-G+1)/(G+1))                                   EUL07710
      SA2=G1*DLOG(((G-1)*W*W+2)/((G+1)*W*W))                                 EUL07720
      SA=SB-SA1-SA2                                                          EUL07730
      QB=(QQB+RRB)/2.0D00                                                    EUL07740
      UB=QB-VS                                                               EUL07750
      UA=UB+AB*DQ                                                            EUL07760
      QA=UA+VS                                                               EUL07770
      QQA=QA+AA*SA                                                           EUL07730
      RRA=QA-AA*SA                                                           EJL07790
C                                                                            EUL07800
      QQ(I2(1)-1)=QQA                                                        EUL07810
      RR(I2(1)-1)=RRA                                                        EUL07820
      S(I2(1)-1)=SA                                                          EUL07830
C                                                                            EJL07840
C                                                                            EJL07850
      END IF                                                                 EUL07860
      RETURN                                                                 EUL07870
      END                                                                    EUL07880
C                                                                            EUL07890
      SUBROUTINE BORDER(JSTOP)                                               EJL07900
C                                                                            EJL07910
CCCCC ********************************************                         EUL07920
CCCCC *                                          *                         EUL07930
CCCCC *     PLOTTING AREA SETUP ROUTINE          *                         EUL07940
CCCCC *                                          *                         EUL07950
CCCCC ********************************************                         EJL07960
C                                                                            EUL07970
      INTEGER I                                                              EUL07980
      REAL XORG(4)/1.75,4.65,1.75,4.65/                                      EUL07990
      REAL YORG(4)/2.75,2.75,5.80,5.80/                                      EUL08000
      REAL YMAX(4)/5.5,5.5,1.5,6.20/                                         EUL08010
      REAL YMIN(4)/0.5,0.5,-0.5,4.90/                                        EUL08020
      DO 70 I=1,4                                                            EUL08030
         CALL PHYSOR(XORG(I),YORG(I))                                        EUL08040
         CALL NOBRDR                                                         EUL08050
         CALL AREA2D(2.4,2.4)                                                EUL08060
         CALL FRAME                                                          EJL08070
         CALL GRAF(0.,'SCALE',1.0,YMIN(I),'SCALE',YMAX(I))                   EJL08080
         CALL ENDGR(0)                                                       EJL08090
   70 CONTINUE                                                               EJL08100
      CALL PHYSOR(1.25,2.25)                                                 EUL08110
      CALL NOBRDR                                                            EJL08120
      CALL AREA2D(6.00,6.50)                                                 EUL08130
      CALL HEADIN('SHOCK TUBE RESULTS$',100,1.7,4)                           EUL08140
      CALL HEADIN('FIRST ORDER     N = 101$',100,1.2,4)                      EUL08150
      CALL HEADIN('DENSITY RATIO = 5     TEMP RATIO = 1$',100,1.0,4)         EUL08160
```

```
      CALL HEADIN('PRESSURE RATIO = 5$',100,1.0,4)                      EUL08170
      CALL GRAF(0.,'SCALE',1.,0.,'SCALE',JSTOP)                         EUL08180
      CALL ENDGR(0)                                                     EUL08190
      RETURN                                                            EUL08200
      END                                                               EUL08210
C                                                                       EUL08220
C                                                                       EUL08230
      SUBROUTINE PLOT(J,JSTOP,N,QQ,RR,S,H,XARRAY,                       EUL08240
     #PARRAY,DARRAY,QARRAY,SARRAY,G,G1,G2)                              EJL08250
C                                                                       EJL08260
CC    ***************************************************               EUL08270
CC    *                                                 *               EJL08280
CC    *         GRAPHICAL  PLOTTING  ROUTINE            *               EJL08290
CC    *                                                 *               EJL08300
CC    ***************************************************               EUL08310
C                                                                       EUL08320
      INTEGER I,N,J,JSTOP,KNT(4)/1,4,6,9/                               EUL08330
      REAL*8 QQ(N),RR(N),S(N),H,G,G1,G2                                 EUL08340
      REAL XARRAY(N),QARRAY(N),PARRAY(N),DARRAY(N)                      EUL08350
      REAL SARRAY(N),TEMP                                               EUL08360
      REAL XORG(4)/1.75,4.65,1.75,4.65/                                 EUL08370
      REAL YORG(4)/2.75,2.75,5.80,5.80/                                 EUL08380
      REAL YMAX(4)/5.5,5.5,1.5,6.20/                                    EJL08390
      REAL YMIN(4)/0.5,0.5,-0.5,4.90/                                   EUL08400
      CHARACTER*4 IYNAM                                                 EJL08410
      DIMENSION IYNAM(13)                                               EUL08420
      DATA IYNAM/'PRES','SURE','$   ','DENS',                           EJL08430
     #'ITY$','VELO','CITY','$   ','MODI','FIED',' ENT','ROPY','$   '/   EJL08440
      G=SNGL(G)                                                         EJL08450
      G1=SNGL(G1)                                                       EUL08460
      G2=SNGL(G2)                                                       EJL08470
      H=SNGL(H)                                                         EJL08480
      DO 85 I=1,N                                                       EJL08490
         QQ(I)=SNGL(QQ(I))                                              EUL08500
         RR(I)=SNGL(RR(I))                                              EJL08510
         S(I)=SNGL(S(I))                                                EJL08520
   85 CONTINUE                                                          EUL08530
      DO 80 I=1,N                                                       EUL08540
      QARRAY(I)=(QQ(I)+RR(I))/2.0                                       EUL08550
      TEMP=(QQ(I)-RR(I))*(QQ(I)-RR(I))/(4*S(I)*S(I))                    EUL08560
      DARRAY(I)=((1/TEMP)*EXP(G*(1-G)*(S(I)-G2)))**(-G1)                EUL08570
      SARRAY(I)=S(I)                                                    EJL08580
      PARRAY(I)=TEMP*DARRAY(I)                                          EJL08590
   80 CONTINUE                                                          EJL08600
      DO 83 I=1,4                                                       EUL08610
         CALL PHYSOR(XORG(I),YORG(I))                                   EUL08620
         CALL AREA2D(2.4,2.4)                                           EUL08630
         CALL XNAME('X',1)                                              EJL08640
```

```
      CALL YNAME(IYNAM(KNT(I)),100)                                     EUL08650
      CALL GRAF(0.,'SCALE',1.0,YMIN(I),'SCALE',YMAX(I))                 EJL08660
      IF (I.EQ.1) CALL CURVE(XARRAY,PARRAY,N,0)                         EUL08670
      IF (I.EQ.2) CALL CURVE(XARRAY,DARRAY,N,0)                         EUL08680
      IF (I.EQ.3) CALL CURVE(XARRAY,QARRAY,N,0)                         EUL08690
      IF (I.EQ.4) CALL CURVE(XARRAY,SARRAY,N,0)                         EUL08700
      CALL ENDGR(0)                                                     EUL08710
   83 CONTINUE                                                          EUL08720
      RETURN                                                            EUL08730
      END                                                               EUL08740
C                                                                       EJL08750
      SUBROUTINE EXACT(N,XINIT,T,VHEAD,VTAIL,VCDE,VSE,DLCD,DLSH,QQ,RR,S,EUL08760
     #H,XARRAY,DARRAY,G,G1,G2,DRI)                                      EJL08770
C                                                                       EUL08780
CCC   ***************************************                           EUL08790
CCC   *                                     *                           EUL08800
CCC   *      EXACT SOLUTION COMPARISON ROUTINE                          EJL08810
CCC   *                                     *                           EUL08820
CCC   ***************************************                           EUL08830
C                                                                       EJL08840
      INTEGER N                                                         EUL08850
      REAL*8 QQ(N),RR(N),S(N),H,G,G1,G2,T,DRI                           EUL08860
      REAL XEXACT(6),YEXACT(6),XINIT,VHEAD,VTAIL,VCDE                   EUL08870
      REAL VSE,DLCD,DLSH,TEMP,XARRAY(N),DARRAY(N)                       EJL08880
C                                                                       EJL08890
      XEXACT(1)=0.0                                                     EJL08900
      XEXACT(2)=XINIT+T*VHEAD                                           EJL08910
      XEXACT(3)=XINIT+T*VTAIL                                           EUL08920
      XEXACT(4)=XINIT+T*VCDE                                            EUL08930
      XEXACT(5)=XINIT+T*VSE                                             EJL08940
      XEXACT(6)=1.0                                                     EJL08950
      YEXACT(1)=SNGL(DRI)                                               EUL08960
      YEXACT(2)=YEXACT(1)                                               EUL08970
      YEXACT(3)=DLCD                                                    EJL08980
      YEXACT(4)=YEXACT(3)                                               EUL08990
      YEXACT(5)=DLSH                                                    EJL09000
      YEXACT(6)=1.0                                                     EUL09010
C                                                                       EUL09020
      DO 90 I=1,N                                                       EJL09030
      QQ(I)=SNGL(QQ(I))                                                 EUL09040
      RR(I)=SNGL(RR(I))                                                 EJL09050
      S(I)=SNGL(S(I))                                                   EUL09060
      TEMP=(QQ(I)-RR(I))*(QQ(I)-RR(I))/(4*S(I)*S(I))                    EUL09070
      DARRAY(I)=((1/TEMP)*EXP(G*(1-G)*(S(I)-G2)))**(-G1)                EJL09080
   90 CONTINUE                                                          EUL09090
      CALL PHYSOR(2.65,2.25)                                            EUL09100
      CALL NOBRDR                                                       EJL09110
      CALL AREA2D(6.75,4.0)                                             EUL09120
```

```
CALL XNAME('X',1)                                                    EUL09130
CALL YNAME('DENSITY',7)                                              EUL09140
CALL HEADIN('DENSITY DISTRIBUTION$',100,1.3,4)                       EUL09150
CALL HEADIN('FIRST ORDER       N = 51$',100,1.0,4)                   EUL09160
CALL HEADIN('DENSITY RATIO = 5    TEMP RATIO = 1$',100,1.0,4)        EUL09170
CALL HEADIN('PRESSURE RATIO = 5$',100,1.0,4)                         EUL09180
CALL LINES('EULER-1 SOLUTION$',IPKRAY,2)                             EUL09190
CALL LINES('EXACT SOLUTION$',IPKRAY,1)                               EUL09200
CALL FRAME                                                           EUL09210
CALL GRAF(0.0,'SCALE',1.0,0.0,'SCALE',6.0)                           EUL09220
CALL MARKER(0)                                                       EUL09230
CALL CURVE(XEXACT,YEXACT,6,-1)                                       EUL09240
CALL LEGLIN                                                          EUL09250
CALL CURVE(XARRAY,DARRAY,N,0)                                        EUL09260
CALL LEGEND(IPKRAY,2,4.25,2.75)                                      EUL09270
CALL ENDGR(0)                                                        EUL09280
RETURN                                                               EUL09290
END                                                                  EUL09300
```

LIST OF REFERENCES

1. Eidelman, S., Colella, P. and Shreeve, R.P., "Application of the Godunov Method and Higher Order Extensions of the Godunov Method to Cascade Flow Modeling," AIAA Journal, Vol. 22, No. 11, pp. 1609-1615, November, 1984.

2. Eidelman, S. and Shreeve, R.P., "Numerical Modeling of the Non-steady Thrust Produced by Intermittent Pressure Rise in a Diverging Channel," ASME Winter Annual Meeting, New Orleans, ASME Bk. No. G00273, December, 1984.

3. Eidelman, S., "The Problem of Gradual Opening in Wave Rotor Passages, AIAA Journal of Propulsion and Power, Vol. 1, No. 1, Jan-Feb., 1985.

4. Naval Postgraduate School Contractor Report NPS67-84-007CR, Development and Evolution of a Numerical Solution of the Euler Equations Using the Godunov Method, by S. Eidelman, November, 1984.

5. Naval Postgraduate School Contractor Report NPS67-85-006CR, Wave Rotor Research: A Computer Code for Preliminary Design of Wave Diagrams, by A. Mathur, June, 1985.

6. McDonnell Aircraft Company Report 83-031, A Natural Formulation for Numerical Solution of the Euler Equations, by A. Verhoff and P.J. O'Neil, 1983.

7. NASA Contractor Report 3712, An Improved Lambda-Scheme for One-Dimensional Flows, by G. Moretti and M.T. DiPiano, 1983.

8. Keenan, J.H., Thermodynamics, John Wiley & Sons, Inc., 1941, p. 88.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	A. Verhoff, Code D341 McDonnell Douglas Corporation Box 516 St. Louis, Missouri 63166-0516	3
4.	F. S. Salacka 7 Woodside Road East Apalachin, New York 13732-9428	3
5.	Office of Research Administration, Code 012 Naval Postgraduate School Monterey, California 93943-5004	1
6.	Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5004	1
7.	Director, Turbopropulsion Laboratory, Code 67Sf Department of Aeronautics Naval Postgraduate School Monterey, California 93943-5004	11
8.	Dr. Gerhard Heiche Naval Air Systems Command, Code 03D Washington, D.C. 20361-0001	1
9.	Mr. George Derderian Naval Air Systems Command, Code 310E Washington, D.C. 20360-0001	1
10.	Dr. M. Keith Ellingsworth Office of Naval Research, Code 1132P 800 North Quincy Street Arlington, Virginia 22217-0001	2

11. Professor Ch. Hirsch 1
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussels
Belgium

12. Mr. P. Tramm 1
Allison Gas Turbine Division
of General Motors
P.O. Box 420
Indianapolis, Indiana 46206-0420

13. Calvin Ball, Samll Gas Turbine Engines 1
NASA Lewis Research Center, MS77-6
21000 Brookpark Road
Cleveland, Ohio 44134-1525

14. David Gordon Wilson 1
M.I.T. Mechanical Engineering, Room 3-455
Cambridge, Massachusetts 02139

15. Helmut E. Weber 1
Professor, Department of
Mechanical Engineering
San Diego State University
San Diego, California 92182-0191

16. Robert Taussig 1
Director, Energy Technology
Spectra Technology, Inc.
2755 Northup Way
Bellevue, Washington 98004-1495

17. Dr. Shmuel Eidelman, Research Physicist 1
Science Application International Co.
1710 Goodridge Dr. Mail Stop G-8-1
McLean, Virginia 22306

215196

Thesis
S15258 Salacka
c.1 Review, implementation and test of the QAZID computational method with a view to wave rotor applications.